

UNIVERSIDADE FEDERAL DO PARANÁ

NICKOLAS MENEZES DA SILVA

REDER WEB: UMA PLATAFORMA WEB PARA ORGANIZAÇÃO E
ANÁLISE DE REDES MODULARES

CURITIBA

2017

NICKOLAS MENEZES DA SILVA

REDER WEB: UMA PLATAFORMA WEB PARA ORGANIZAÇÃO E
ANÁLISE DE REDES MODULARES

Dissertação apresentada como requisito parcial à
obtenção do grau de Mestre em Bioinformática, no
Curso de Pós-Graduação em Bioinformática, Setor
de Educação Profissional e Tecnológica, da
Universidade Federal do Paraná.

Orientador: Prof. Dr. Mauro Antonio Alves Castro
Coorientador: Prof. Dr. Alessandro Braweman

CURITIBA

2017

Catálogo na Publicação

Sistema de Bibliotecas UFPR

Biblioteca de Educação Profissional e Tecnológica

S586	<p>Silva, Nickolas Menezes da</p> <p>RedeR Web: uma plataforma Web para organização e análise de redes modulares. - Curitiba, 2017.</p> <p>64 f.; il.: tab.</p> <p>Orientador: Mauro Antonio Alves Castro</p> <p>Coorientador: Alessandro Braweman</p> <p>Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Educação Profissional e Tecnológica, Curso de Pós-Graduação em Bioinformática.</p> <p>Inclui Bibliografia.</p> <p>1. Sistemas biológicos. 2. Visualização da informação. 3. Teoria dos grafos – Processamento de dados. 4. Redes modulares. 5. Bioinformática. I. Castro, Mauro Antonio Alves. II. Braweman, Alessandro. III. Título. IV. Universidade Federal do Paraná.</p> <p>CDD 005.369</p>
------	--



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

Pós-Graduação em Bioinformática WWW.BIOINFO.UFPR.BR
E-mail: bioinfo@ufpr.br Tel: 41 33614906

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em BIOINFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **NICKOLAS MENEZES DA SILVA** intitulada: "**RedeR WEB: uma plataforma web para organização e análise de redes modulares**", após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 29 de Setembro de 2017.



Dr. Mauro Antonio Alves Castro

Presidente

Programa de Pós-graduação em Bioinformática – UFPR



Dr. Roberto Tadeu Raittz

Avaliador Interno

Programa de Pós-graduação em Bioinformática – UFPR



Dr.ª Vania Aparecida Vicente

Avaliadora Externa

Programa de Pós-Graduação de Engenharia de Bioprocessos e Biotecnologia – UFPR

AGRADECIMENTOS

Em primeiro lugar, agradeço à Deus por permitir que mais uma etapa importante da minha vida seja concretizada.

Aos meus orientadores Prof. Dr. Mauro Antonio Alves Castro e Prof. Dr. Alessandro Brawerman por todo apoio e cooperação no desenvolvimento deste projeto.

A minha amada esposa, Karine M. L. Menezes por meu auxiliar, suportar e embelezar meus dias. E também por neste período do mestrado me presentear com nosso amado filho, o Bernardo.

A toda minha família por me incentivar, apoiar e participar ativamente desta conquista.

A todos os docentes, técnicos e discentes do Programa de Pós-Graduação em Bioinformática.

A Pró-Reitoria de Pesquisa e Pós-Graduação gestão (2013-2016) e gestão atual pelo incentivo e suporte financeiro.

Ao servidor José Henrique Ferreira Pinto, coordenador da equipe SIGA, por toda ajuda, incentivo e experiência proporcionada cotidianamente.

A toda Equipe SIGA pela inestimável contribuição técnica, científica e emocional no planejamento e execução deste projeto.

“Embora a verdade nem sempre seja humildade, a
humildade é sempre verdade: o reconhecimento sem
rodeios de que devo minha vida, meu ser e minha salvação
a outro. Esse ato fundamental jaz no âmago de nossa
reação à graça ”

Brennan Manning

RESUMO

A visualização de grafos ou redes tem como principal objetivo auxiliar o processo de análise e compreensão de conjuntos de dados complexos, proporcionando perspectivas diferenciadas da mesma informação, potencializando a apropriação do conhecimento pelo usuário. Tratando-se de sistemas biológicos, a visualização e análise de redes moleculares têm sido centrais para o entendimento desses sistemas. No entanto, ainda há uma grande lacuna tecnológica, pois a maioria dos softwares de visualização de redes existentes não tem tirado proveito da modularidade e hierarquia características de sistemas biológicos, principalmente quando essas sistemas possuem diversos níveis hierárquicos. A consequência disso é que muitas vezes a visualização da informação acaba sendo comprometida, dificultando a interpretação dos dados. Motivado pelos atributos do software RedeR (Castro et al., 2012), um pacote R/Bioconductor combinado com uma aplicação Java SE, apresentamos a versão RedeR Web, uma plataforma Web, intuitiva, responsiva que visa a simplificação da visualização de redes modulares. Neste estudo mostraremos como essa nova plataforma obtém proveito da modularidade e da hierarquia das redes biológicas, como otimizamos o tempo de renderização dos elementos, bem como apresentaremos um estudo de caso para demonstrando como podemos usar esta ferramenta para visualizar a estrutura hierárquica de termos do GeneOntology.

Palavras-chave: grafos, visualização, redes, sistemas biológicos, modularidade, hierarquia.

ABSTRACT

Visualization of graphs or networks has as main objective to assist the process of analysis and understanding of complex data sets, providing different perspectives of the same information, enhancing the appropriation of knowledge by the user. In the case of biological systems, the visualization and analysis of molecular networks have been central to the understanding of these systems. However, there is still a large technological gap, since most of the existing visualization software networks have not taken advantage of the modularity and hierarchy characteristics of biological systems, especially when these systems have several hierarchical levels. The consequence of this is that often the information visualization ends up being compromised, making hard to interpret the data. Motivated by the attributes of RedeR software (Castro et al., 2012), a R/Bioconductor package combined with a Java SE application, we show the RedeR Web version, an intuitive, responsive Web platform aimed at simplifying the visualization of modular networks. In this study we will show how this new platform takes advantage of the modularity and hierarchy of biological networks, how we optimize the rendering time of the elements, and we will present a case study to demonstrate how we can use this tool to visualize the hierarchical structure of terms in GeneOntology.

Keywords: graphs, visualization, networks, biological systems, modularity, hierarchy.

SUMÁRIO

APRESENTAÇÃO	12
CAPITULO 1: CONCEITOS FUNDAMENTAIS	13
1. INTRODUÇÃO	14
2. OBJETIVOS.....	15
2.1 OBJETIVO GERAL	15
2.2 OBJETIVOS ESPECÍFICOS.....	15
3. REVISÃO DE LITERATURA.....	15
3.1 BIG DATA	15
3.2 VISUALIZAÇÃO DA INFORMAÇÃO	18
3.3 HISTÓRIA E ORIGEM DA VISUALIZAÇÃO	18
3.4 COGNIÇÃO E A VISUALIZAÇÃO DA INFORMAÇÃO	20
3.5 CONCEITO DA VISUALIZAÇÃO DA INFORMAÇÃO	21
3.6 ETAPAS DA VISUALIZAÇÃO DA INFORMAÇÃO	22
3.7 DIMENSIONALIDADE	23
3.8 MÉTODOS DE VISUALIZAÇÃO DA INFORMAÇÃO.....	23
3.9 TÉCNICAS DE INTERATIVIDADE E CONTROLES DE PONTO DE VISTA	26
3.10 ESTUDO DE REDES E TEORIA DE GRAFOS.....	27
3.11 TEORIA DAS REDES COMPLEXAS.....	29
3.11.1 COEFICIENTE DE CLUSTERIZAÇÃO E DISTRIBUIÇÃO DE GRAUS.....	29
3.11.2 MODELOS DE REDES.....	30
3.12 REDES BIOLÓGICAS	31
CAPITULO 2: DESENVOLVIMENTO DA PLATAFORMA.....	33
1. INTRODUÇÃO	34
2. METODOLOGIA E MODELAGEM	34
3. PLANO DE ATIVIDADES	35
4. PLANO DE RISCOS.....	38
5. TECNOLOGIAS.....	39
6. INFRAESTRUTURA.....	41
7. FORMATOS DE ARQUIVOS	41
8. RESULTADOS E APRESENTAÇÃO DA PLATAFORMA	43
9. TESTES.....	52
CAPITULO 3: ESTUDO DE CASO	55
1. INTRODUÇÃO	56
2. POR QUÊ UM ESTUDO DE CASO COM O GENE ONTOLOGY?.....	56
3. O ESTUDO DE CASO.....	57
4. RESULTADOS E DISCUSSÃO.....	59
5. CONSIDERAÇÕES FINAIS E PERSPECTIVAS FUTURAS	59
REFERÊNCIAS.....	60

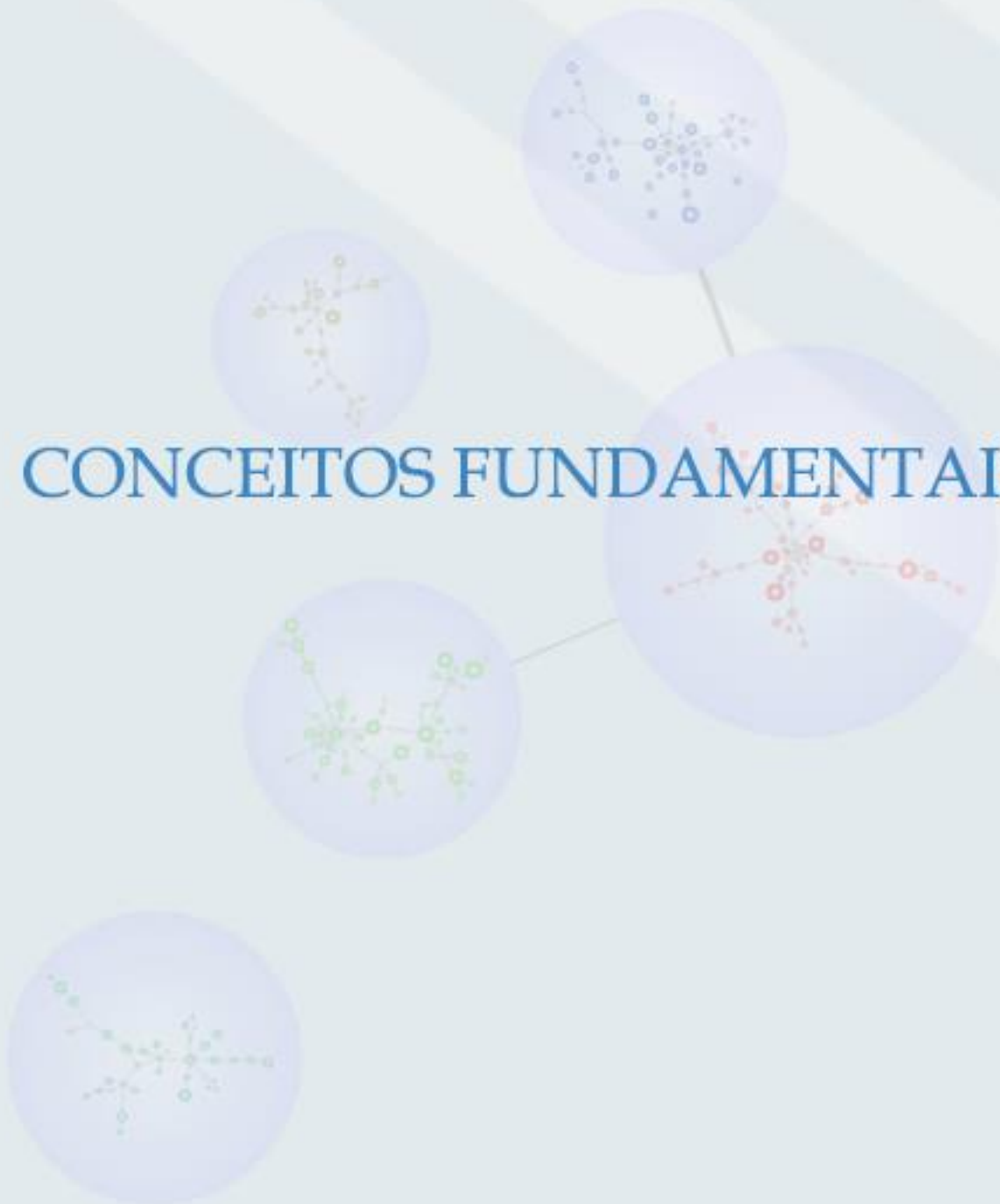
APRESENTAÇÃO

Esta dissertação foi organizada em três capítulos, seguindo as diretrizes estabelecidas pela Universidade Federal do Paraná. O capítulo 1 apresenta um levantamento dos conceitos que fundamentam o desenvolvimento do trabalho. No capítulo 2 foi apresentado o desenvolvimento da plataforma web contendo a descrição do desenvolvimento do trabalho, metodologias utilizadas e as linguagens de programação, bibliotecas e empregadas na criação da plataforma proposta.

O Capítulo 3 apresenta a aplicação do software desenvolvido, com base em um estudo de caso, discussão e as considerações do trabalho realizado.

Capítulo 1

CONCEITOS FUNDAMENTAIS



CAPÍTULO 1: CONCEITOS FUNDAMENTAIS

1- INTRODUÇÃO

Os avanços tecnológicos e computacionais acabam refletindo nas mais diversas áreas de conhecimento, possibilitando uma maior capacidade de armazenamento e manipulação de dados. Na Era do “*Data Never Sleeps*” (Josh James, 2015) tem-se uma crescente e constante produção de dados brutos que poderão ser extremamente valiosos se visualizados na perspectiva adequada.

Na área das ciências biológicas não tem sido diferente. Milhões e milhões de dados são gerados e armazenados em banco de dados, como por exemplo, no GenBank, desenvolvido para armazenar dados de sequências genéticas, no qual estima-se que o número de bases nitrogenadas que dobra a cada 18 meses, tendo hoje cerca de 220.731.315.250 bases depositadas. Esta quantidade gigantesca de dados, juntamente com a sua devida complexidade, origina situações extremamente desafiadoras no que se refere as análises destas informações.

Dentro deste cenário, tem crescido a busca por métodos e técnicas que possam tornar a “*Big Data*” mais compreensível, surgindo então a visualização da informação, área que visa tanto a criação quanto a representação visual de dados, buscando atingir o objetivo de comunicar a informação ao usuário de forma clara e eficiente através de grafos, gráficos, diagramas e outros componentes visuais.

A análise e visualização de redes biológicas é um fator central para o entendimento de sistemas biológicos, possuindo, além da vasta produção de informações, uma lacuna tecnológica, tendo em vista a falta de softwares capazes de permitir ao usuário uma melhor forma de visualização para obter real proveito do significado e valor desses dados.

Há inúmeros softwares desenvolvidos que permitem a visualização de redes biológicas, entre eles o conceituado Cytoscape (<http://www.cytoscape.org/>). Entretanto, a grande maioria trabalha com as redes planas, e isso pode tornar-se um grande problema quando trabalhamos com redes de grande porte, pois são geradas visualizações difíceis de se extrair algum conhecimento, deixando de lado a “pedra angular” dos sistemas biológicos, que, na maioria das vezes, são hierárquicas e modulares. (Barabasi et al., 2004)

Em 2012, surgiu o RedeR (Castro et al. 2012), um software capaz de permitir a análise e visualização interativa de múltiplos níveis de hierarquias, organizando as redes modularmente. Sendo assim esta dissertação propõe levar esta abordagem única empacotada pelo RedeR, que desde sua criação tem obtido vários adeptos, para o ambiente Web, um campo ainda mais desafiador para a análise e visualização de redes.

Mesmo com a existência de vários softwares para análise de redes, nenhum possui a abordagem de organização como o RedeR, tanto em softwares para área de trabalho como em plataformas na Web. Na Web, há uma grande carência por uma plataforma consolidada de análise de redes modulares e hierárquicas.

O RedeR Desktop tem tido um grande número de usuários no nicho em que está enquadrado. No entanto, ao oferecer uma versão para a Web, o

RedeR Web proporcionará maior visibilidade, trazendo maiores adeptos e, consequentemente, ampliará a comunidade que necessita realizar análise e visualização de redes.

2. OBJETIVOS

2.1 OBJETIVO GERAL

O objetivo geral desta dissertação foi criar uma plataforma Web baseada no RedeR que seja intuitiva, visando a simplificação da visualização de redes e a consolidação da plataforma como referência em análise de redes modulares e hierárquicas.

2.2 OBJETIVOS ESPECÍFICOS

Para o desenvolvimento desta plataforma foi necessário atingir os seguintes objetivos específicos:

- a) Buscar a resolução de problemas, como de tempo de renderização e limitação de tamanho das redes.
- b) Proporcionar interface intuitiva e responsiva, visando que o usuário tenha uma experiência facilitada com a plataforma, e possa, sem ter conhecimento de programação, visualizar e analisar seus dados.
- c) Permitir a análise e visualização de redes modulares em dispositivos móveis.
- d) Proporcionar maior visibilidade ao RedeR.

3. REVISÃO DA LITERATURA

3.1 BIG DATA

Nos últimos anos, pode-se observar uma crescente explosão no volume de dados brutos produzidos. Segundo Josh James (2016), estamos na Era do “*Data Never Sleeps*”. Atualmente, a população global que utiliza a Internet cresceu mais de 60%, há mais dispositivos móveis do que pessoas no mundo e essas conexões geram cerca de 18 milhões de megabytes de dados por minuto.

Mesmo com o avanço do poder de armazenamento e processamento dos computadores atuais, existem dados que não se encaixam nos padrões comumente conhecidos. Edd Dumbill (2013) define a *Big Data* (“Mega dados”, na língua portuguesa) da seguinte maneira: “*Big Data* são dados que excedem as capacidades convencionais. Os dados são muito grandes, movimentam-se rapidamente ou não se enquadram nas arquiteturas de banco de dados. Para obter ganho desses dados, é necessário escolher uma maneira alternativa para processá-lo.”

O analista Douglas Laney em 2001, introduziu o primeiro modelo, visando caracterizar a *Big Data*, sendo este intitulado “*Three V’s*”, que inicialmente foi constituído da seguinte maneira:

- 1- Volume: Refere-se à quantidade de dados gerados e armazenados.
- 2- Velocidade: Refere-se à velocidade que os dados são processados.
- 3- Variedade: Refere-se aos vários tipos de dados.

A IBM publicou o modelo “4V’s”, adicionando variabilidade, referindo-se à propriedade dos dados muitas vezes serem inconsistentes, dificultando o gerenciamento e o tratamento destes. E em 2013, Yuri Demchenko propôs uma nova definição com “5V’s”, desta vez, levando em consideração a capacidade de se atribuir valor aos dados.

Nos últimos anos, mais “V’s” têm sido frequentemente associados as definições de *Big Data*. Segundo Rajkumar Buyya (2016) isso demonstra principalmente a tentativa de descrever o aspecto dos dados.

No campo da ciência, o crescimento do uso de *Big Data* não tem sido diferente, especialmente na área das Ciências Biológicas. De acordo com Vivien Marx (2013), cada vez mais biólogos e usuarios de áreas afins têm se juntado ao “*Big-Data Club*”, pois com o avanço da genômica, área específica que estuda o genoma completo de um organismo, mais pesquisadores estão trabalhando com dados densos, encarando desafios com o manuseio, processamento e, por vezes, manipulando informações que outrora pertenciam a outras áreas das ciências.

De acordo com informações disponibilizadas no site do EBI (European Bioinformatics Institute), um dos maiores bancos de dados biológicos do mundo, há mais de 20 *petabytes* de dados sobre genes, proteínas e moléculas depositados em suas bases de dados, tendo poder de armazenamento de mais de 80 *petabytes*.

Segundo Philip Chen (2014) a *Big Data* traz consigo muitas oportunidades atraentes, entretanto, gera inúmeros desafios para a manipulação dos processos que a compõe, como a captura dos dados, o armazenamento, a pesquisa, compartilhamento, análise e visualização (Figura 1).

Para a descoberta do conhecimento e para ornamentar a tomada de decisão, torna-se necessário o desenvolvimento de novas técnicas e tecnologias para analisar a *Big Data*. Estas técnicas reúnem inúmeras disciplinas como mineração de dados, aprendizagem de máquina, redes neurais, reconhecimento de padrões, métodos estatísticos, visualização da informação e análise de redes. (CHEN, P., 2014)



Figura 1 - Processo de análise da Big Data até descoberta do conhecimento
Fonte: CHEN, P. 2014, adaptado.

Pode-se concluir que a *Big Data* não é uma única tecnologia, mas uma combinação de técnicas antigas e novas (Figura 2), que auxilia no ganho de “*insights*” para as aplicações e tomadas de decisões. (HURWITZ et. al, 2013)

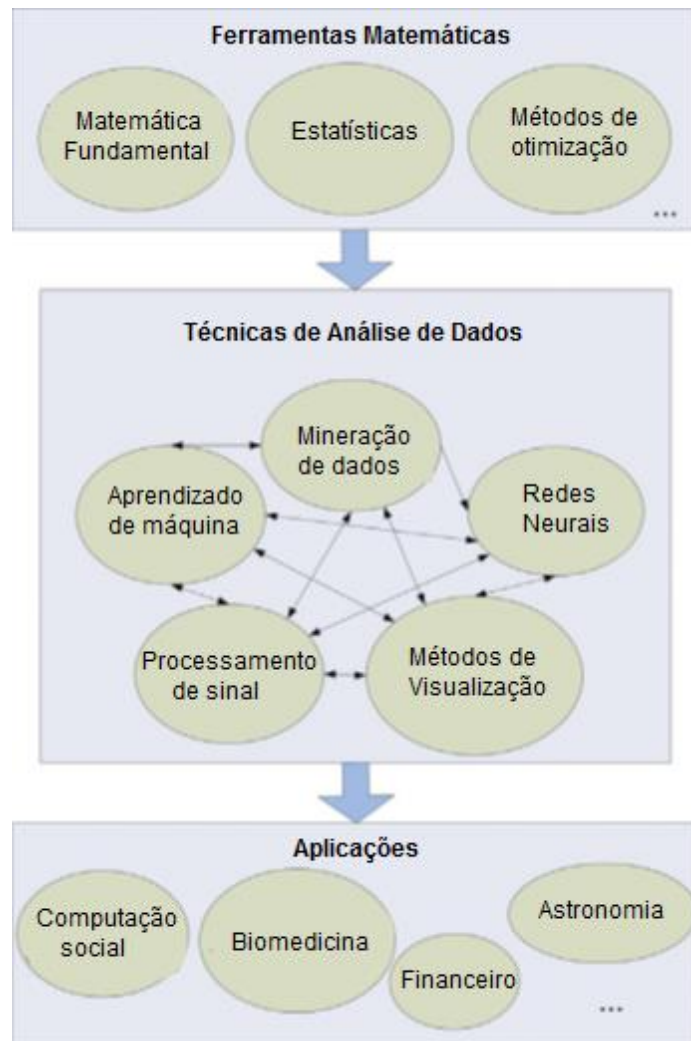


Figura 2 - Técnicas e Tecnologias da *Big Data*
 Fonte: CHEN, P. 2014, adaptado.

3.2 VISUALIZAÇÃO DA INFORMAÇÃO

A visualização da informação é uma nova disciplina que utiliza tecnologias da computação gráfica e baseia-se em conhecimentos da estatística, informática, geometria e psicologia, tendo como princípio oferecer técnicas para criar imagens, gráficos e animações para comunicar determinada informação. (Kmeťová, M., 2010)

3.3 HISTÓRIA E ORIGEM DA VISUALIZAÇÃO

Desde os primórdios da humanidade, observa-se fatos que descrevem que a mente humana sempre foi muito visual. Na pré-história, quando os seres humanos ainda se esforçavam para manter a própria sobrevivência, eles já descreviam suas estratégias de caça e estatísticas nas paredes de cavernas. (Aparicio, M., 2014)

No entanto, os grandes avanços que marcam a origem da visualização começaram a partir da expansão marítima, no qual os primeiros mapas e

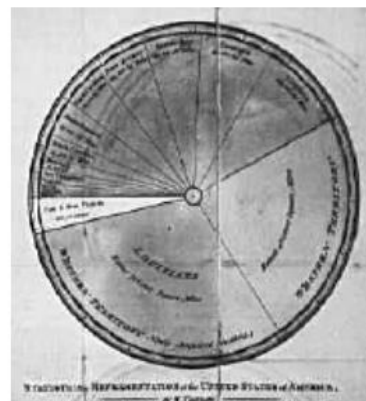
diagramas geométricos foram criados para auxiliar a navegação. (Michael Friendly, 2009)

No século XVI, surgiu uma grande necessidade de medir atributos físicos, o que ocasionou a geração de novas técnicas e instrumentos para a observação e representação destas medidas (FRIENDLY, M., 2009). No século XVII ascendeu-se a geometria analítica, teoria da probabilidade, estatística demográfica, aritmética e outras teorias que desenvolveram o chamado “pensamento estatístico”. Desde então, novas formas gráficas passaram a ser criadas para auxiliar provas matemáticas, funções e cálculos (FRIENDLY, M., 2009).

Com o surgimento da infografia moderna e o uso de sistemas computacionais, novos conceitos de manipulação de dados, técnicas gráficas e visualizações multidimensionais foram empregadas (Figuras 3 e 4) promovendo a evolução da representação do conhecimento (FRIENDLY, M., 2009).



A



B

Figura 3- Visualizações multidimensionais empregadas e a evolução da representação do conhecimento: A- Possível primeiro Mapa-Mundi, descrito nos livros de Heródoto, na Grécia; B - Início da infografia moderna: Gráfico de pizza circular, de William Playfair (1759-1823), Inglaterra. Fonte: Milestones in the History of Data Visualization. Fonte: Milestones in the History of Data Visualization.



Figura 4 – Previsão do Tempo de Jornal Americano desenvolvido por George Rorick, 1982.
Fonte: Milestones in the History of Data Visualization.

3.4 COGNIÇÃO E A VISUALIZAÇÃO DA INFORMAÇÃO

O sistema cognitivo humano, em tarefas simples, desempenha apenas funções conectoras entre o sistema perceptivo e os sistemas motores. No entanto, a maioria das tarefas executadas por um indivíduo são complexas e abrangem aprendizagem, recuperação de fatos e soluções de problemas (CARD, S. et al., 1983).

Tor Norretranders (1999), um físico dinamarquês, realizou uma pesquisa sobre a velocidade dos sentidos em comparação com termos computacionais, criando um conceito denominado "*Bandwidth of Senses*" (Largura de banda dos sentidos: Figura 5), no qual observa-se que a visão é o sentido mais poderoso, sendo sua velocidade de transferência igual a 1.250 mb/s equivalente a uma rede de computadores. O toque vem em segundo lugar, com largura de banda 10 vezes menor que a visão, sendo comparado a uma porta USB.

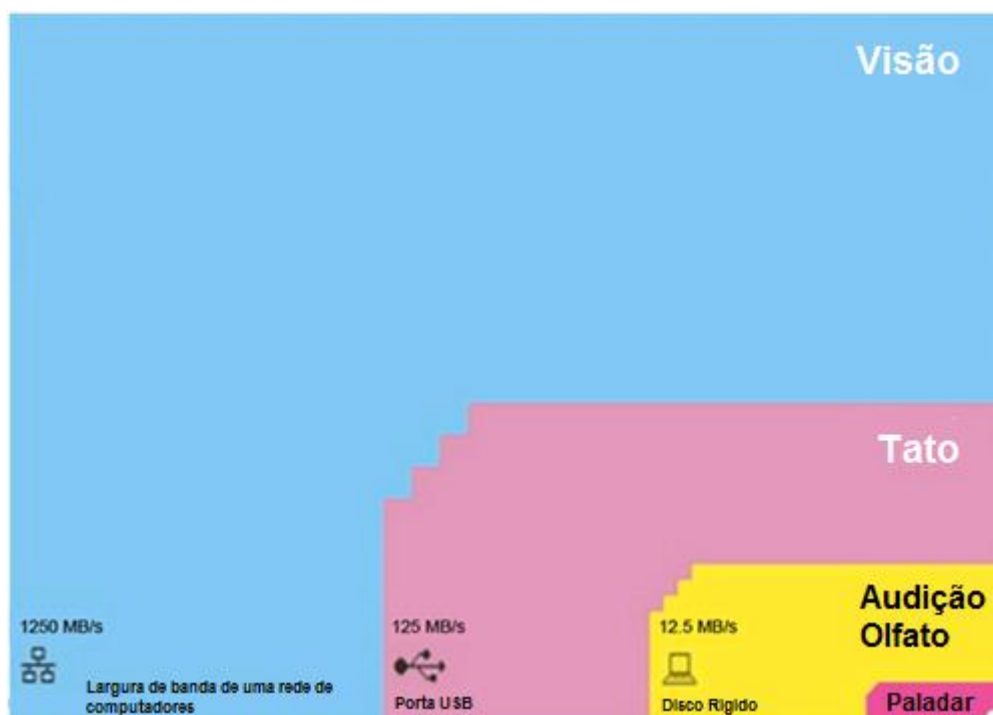


Figura 5 – Largura de banda dos sentidos: “*Bandwidth of Senses*” Adaptado de Tor Norretranders, 2015.

Portanto, as visualizações de informações têm se tornado cada vez mais importantes para os sistemas cognitivos, pois adquirimos mais informações através da visão, do que até mesmo de outros sentidos combinados (Colin Ware, 2013).

3.5 CONCEITO DA VISUALIZAÇÃO DA INFORMAÇÃO

A visualização da informação (VI) é a representação visual de espaços e estruturas de informações abstratas, conjuntamente com as suas respectivas interações de modo a facilitar a assimilação e compreensão dos dados (KEITH ANDREWS, 2016).

De acordo com Muzammil Khan (2011) com a quantidade de dados que têm sido gerados das mais variadas fontes e formatos, encontramos uma questão principal: como usar a visualização da informação para extrair conhecimento e informação útil de dados brutos armazenados. Colin Ware (2013) também afirma que a questão crítica da VI é como transformar os dados de uma maneira que as pessoas possam entendê-los para uma tomada de decisão.

Não há nenhuma fórmula mágica para construir uma representação gráfica eficiente a partir de dados brutos. Tudo depende da natureza dos dados, do tipo de informação com seu devido uso e, principalmente, da habilidade criativa de quem produzirá a interface gráfica. (Riccardo Mazza, 2004). Entretanto, Edward Tufte (1983) expressa que, para alcançar “excelência gráfica”, é necessário comunicar ideias complexas com clareza, precisão e eficiência, descrevendo que o elemento gráfico deve:

- Mostrar a informação.
- Induzir o espectador a pensar sobre a informação que está sendo exibida, e não sobre a metodologia, o design ou a tecnologia na produção gráfica.
- Evitar a distorção do que as informações têm a dizer.
- Tornar os grandes conjuntos de dados coerentes.
- Encorajar a visão do usuário a olhar para diferentes partes dos dados.
- Permitir que os dados sejam vistos em diversos níveis de detalhes, desde uma visão geral a uma estrutura bem específica.

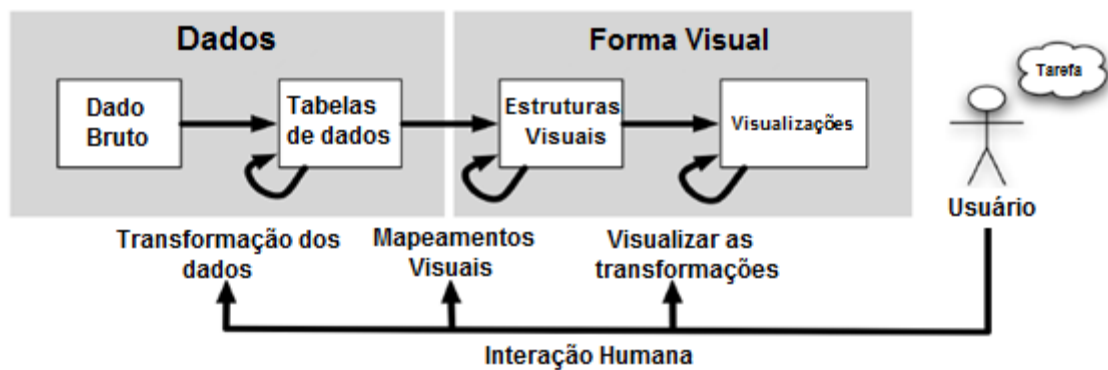
Seguindo estes princípios, e conforme Colin Ware (2013), conclui-se que a visualização da informação quando bem utilizada, fornece a capacidade de compreender grandes quantidades de dados, permite a percepção de propriedades não antecipadas, pode frequentemente ser utilizada como um controle de qualidade dos dados, permite a percepção de padrões, assim como pode atuar na formação de hipóteses.

Outros benefícios podem ser atribuídos ao responder a questão de “O porquê visualizar dados?”. Wolfgang Aigner (2010) diz que a VI reduz a pesquisa, pois uma grande quantidade de dados é exibida em um espaço pequeno, facilitando a percepção de relações que são mais difíceis de induzir de outras formas, assim como permitindo fornecer meios manipuláveis para navegar na informação, diferente de diagramas estáticos.

3.6 ETAPAS DA VISUALIZAÇÃO DA INFORMAÇÃO

Colin Ware (2013) divide a visualização da informação em quatro etapas, que obtêm feedbacks entre si, as quais são 1 A captura e armazenamento dos dados brutos; 2. O pré-processamento: Transformar os dados para facilitar sua manipulação; 3. Mapeamento dos dados selecionados que é realizado por algoritmos computacionais para formar a imagem na tela. Sendo que, nesta etapa o usuário pode mudar mapeamentos, realçar subconjuntos ou mudar o tipo de visão; 4.O sistema perceptivo e cognitivo humano.

Essas etapas podem ser observadas também no modelo de referência descrito por Stuart Card (1999) para a visualização da informação (Figura 6). Observa-se os dados brutos, as transformações dos dados, os mapeamentos visuais e as interações humanas realizadas pelo usuário, afim de executar suas tarefas.



Dado Bruto: Formatos idiossincrático

Transformação dos dados: mapear e organizar dados brutos de forma apropriada para visualização

Tabelas de dados: Relações e metadados

Mapeamentos Visuais: Codificar dados abstratos em uma representação visual

Estruturas Visuais: substratos espaciais + marcas + propriedades gráficas

Transformações Visuais: Alterar a visão ou perspectiva sobre a representação visual

Visualizações: Parâmetros gráficos (posição, escala, recorte)

Interação Humana: Influência do usuário em qualquer nível

Figura 6 – Modelo de referência para visualização da informação de Stuart Card

Fonte: Information Visualization, 1999, p. 17, adaptado.

3.7 DIMENSIONALIDADE

Atualmente, nem sempre podemos entender quando o uso do 3D é mais vantajoso que o 2D. Com os futuros avanços tecnológicos, as visualizações tridimensionais passarão a ser mais sofisticadas, com isso será importante que estes avanços sejam incorporados aos softwares de visualização (NAHUM GERSHON et. al, 1998).

Jakob Nielsen (1998) afirma que o uso do 3D em computadores enfrenta as seguintes dificuldades: A tela e o mouse são dispositivos 2D, desta maneira, não se alcança o 3D verdadeiro; Controlar um ambiente 3D com as atuais técnicas de interação como “*dragging*”, “*scrolling*”, torna-se uma tarefa custosa, pois foram desenvolvidas para a manipulação 2D e o usuário pode acabar perdendo o foco de sua tarefa principal, pois precisa usar muito de sua atenção à navegação tridimensional.

Portanto, quanto mais simples as visualizações forem, mais fáceis serão de entender. Deste modo, deve-se explorar primeiramente o uso do 2D, sendo este suficiente para análise visual de dados (MIKSCH, S., 2014).

3.8 MÉTODOS DE VISUALIZACAO DA INFORMACÃO

De acordo com Muzammil Khan (2011), os métodos de visualização da informação devem possuir uma interface interativa de dados para amplificar a capacidade cognitiva e perceptiva. Deve-se permitir que a imagem visualizada seja versátil, afim de que o usuário possa interagir com a informação.

Um dos métodos mais comuns de visualização de dados é o gráfico de barras, conhecido também como gráfico de colunas. Geralmente é utilizado para representar dados discretos (Figura 7). O controle do gráfico é dado pela barra horizontal e na barra vertical encontram-se os valores (KHAN, M., 2011).

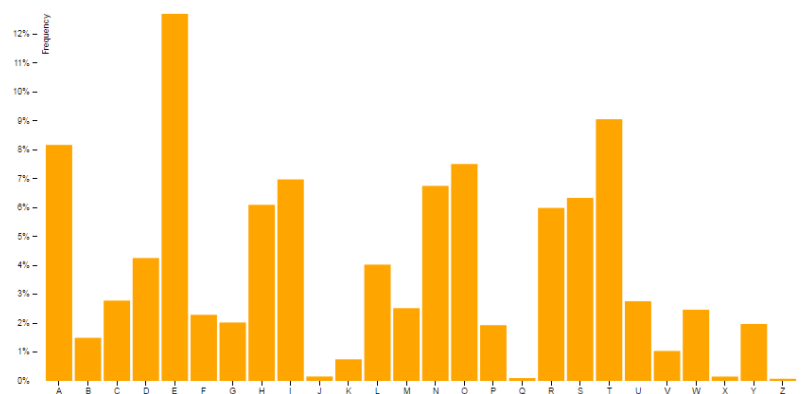


Figura 7 – Exemplo de Gráfico de Barras

Fonte: D3.js Gallery (Disponível em: <https://github.com/d3/d3/wiki/Gallery>)

Considerado uma das formas mais poderosas de representação para análise de dados é o diagrama de dispersão denominado de *scatterplot* (figura 8) que permite que ao parrear duas variáveis sejam medidos a quantidade de associação entre elas, a dependência de uma em outra, a visualização de clusters de pontos, *outliers* e outras informações. (Anscombe 1973; Chambers et al. 1983)

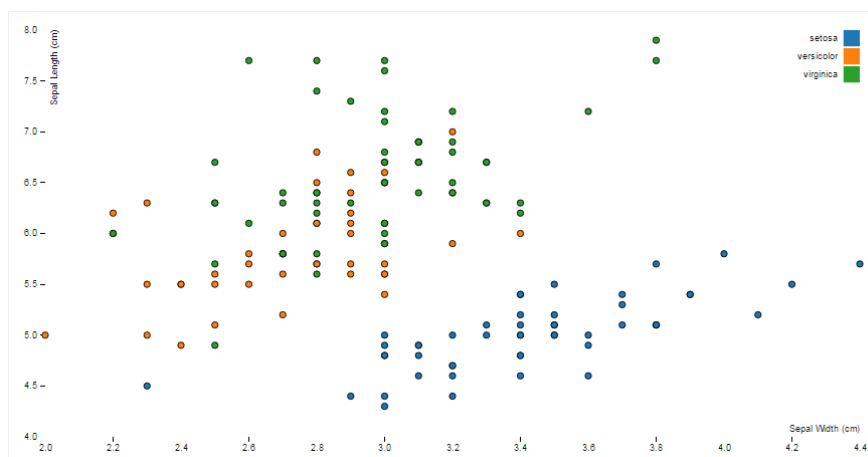


Figura 8 - Exemplo de diagrama de dispersão “*ScatterPlot*”

Fonte: D3.js Gallery (Disponível em: <https://github.com/d3/d3/wiki/Gallery>)

O *tree map* é um método para a visualização de estruturas hierárquicas, que utiliza eficientemente o espaço de exibição disponível, mapeando hierarquias em retângulos aninhados (Figura 9). Com isso, torna-se possível ao usuário visualmente analisar os resultados esperados, reconhecer padrões, comparar nós e subnós em várias profundidades diferentes (JOHNSON, B. & SHNEIDERMAN, 1991).

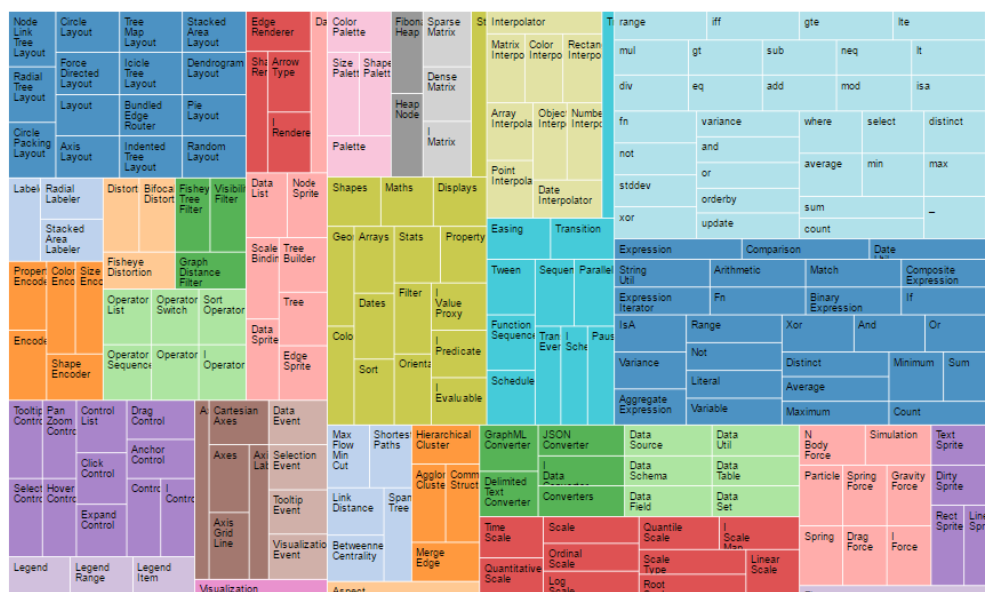


Figura 9 – Exemplo de Tree Map

Fonte: D3.js Gallery (Disponível em: <https://github.com/d3/d3/wiki/Gallery>)

O *circle packing* (Figura 10) é um método muito similar ao *tree map*. Entretanto, utiliza círculos ao invés de retângulos. Sendo assim proporciona maior facilidade na visualização de agrupamentos e relações estruturais. (Wang, W. et Al, 2006)

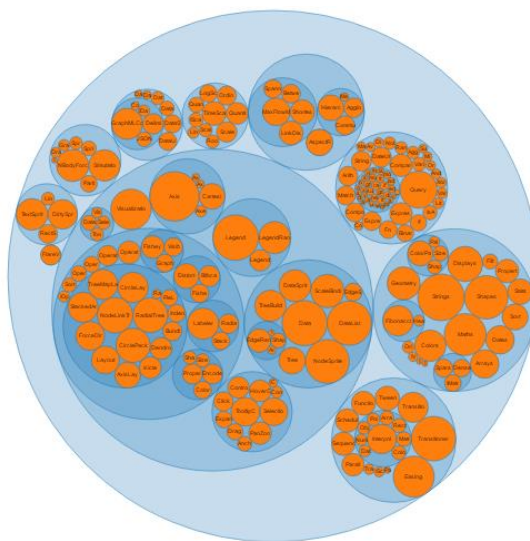


Figura 10 – Exemplo de Circle Packing

Fonte: D3.js Gallery (Disponível em: <https://github.com/d3/d3/wiki/Gallery>)

O diagrama de redes, também conhecido como grafo, é formado por duas entidades principais, os “nós” e as “arestas”, sendo que o relacionamento entre elas constitui a topologia da rede. Outros atributos também podem ser adicionados em diferentes níveis de detalhe. As tarefas executadas com a análise de rede são das mais variadas, como entendimento de propriedades estatísticas de topologias, identificação de nós significativos, detecção de anomalias, entre outros. (Heymann S., 2006)

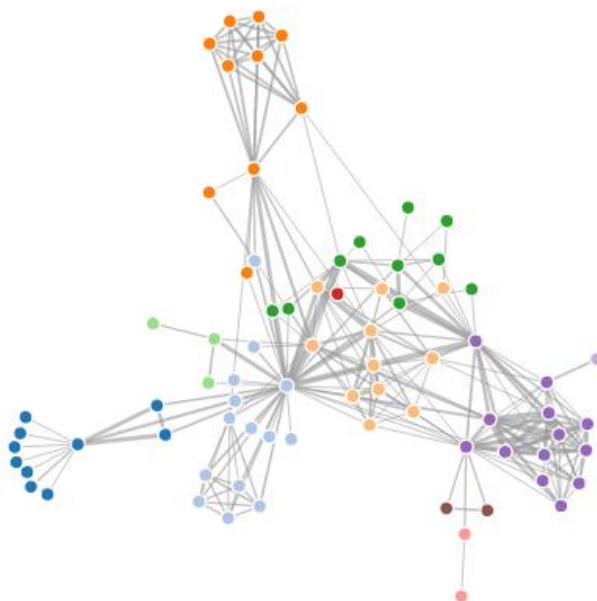


Figura 11 – Exemplo de Network Analysis

Fonte: D3.js Gallery (Disponível em: <https://github.com/d3/d3/wiki/Gallery>)

3.9 TÉCNICAS DE INTERATIVIDADE E CONTROLES DE PONTO DE VISTA

Um problema comum na Visualização da informação (VI) é a representação de um grande lote de dados em uma pequena área de exibição. Há várias técnicas propostas para solucionar este problema (SPENCE, 2001).

Com as transformações de visão e controle de ponto de vista é possível aumentar o nível de informação que poderá ser visualizada e tornar os detalhes mais visíveis. A interatividade do usuário com essas técnicas permite que seja delimitado um espaço de trabalho para a produção de senso visual (CARD ET. AL., 1999). Frequentemente o usuário necessita ter uma visão mais próxima ao elemento visualizado, com o intuito de obter uma maior clareza ou detalhamento de determinada informação. Com isso, a técnica de zoom permite que seja alterada a escala da interface em diferentes níveis de detalhe (KHAN, M., 2011).

Uma segunda técnica conhecida como PAN permite que o usuário deslize uma janela sobre o conteúdo total, sendo que, nesta janela apenas parte da informação é exibida. Quando o usuário move o mouse até o limite da janela, esta também é movida na mesma proporção. Robert Spence (2001) define a técnica de *panning* como sendo um movimento suave de uma janela de visualização sobre uma imagem de maior tamanho.

A combinação das técnicas de *pan* e *zoom* possibilita ter uma escalabilidade ilimitada da interface. O usuário pode aplicar um zoom a um nível de detalhe e deslizar sob o resto da interface, sem que tenha que voltar a escala do zoom e aplicá-lo novamente sobre outra região. (Khan, M., 2011)

Além disso tem-se a técnica de *scrolling* também denominada de SCROL que consiste no movimento de dados sob uma janela, que tem a capacidade de exibir apenas uma fração de todo o conteúdo (SPENCE, 2001).

Uma outra estratégia, reconhecida como FOCUS + CONTEXT aumenta uma área de foco diretamente no contexto global. O usuário facilmente desliza o foco sob a visão geral, obtendo maiores detalhes. Para que isso seja possível, é necessário distorcer a interface global, por isso essa técnica é conhecida também como *fish-eye view* ou *distortion-oriented*. (KHAN, M, 2011; BEDERSON, B.B et. al, 1996; LEUNG, Y. K & APPERLEY, M. D., 1994).

Também existe disponível o uso das técnicas de filtro (FILTER) que consiste em remover elementos desinteressantes. Removendo partes da visualização, o contexto global será alterado, por isso é importante verificar se existe a possibilidade de abstrair a informação que será removida. (Maletic et. al., 2002). Uma outra opção é técnica de “*overview + detail*” constituída de duas janelas. A primeira com uma visão geral da interface visual e a segunda possuindo um enfoque amplificado em uma determinada região (CARD et. al., 1999).

Da mesma forma a técnica denominada HISTORY consiste em manter um histórico das ações realizadas, dando suporte para que o usuário possa avançar ou retroceder em etapas dentro de sua análise ou construir um caminho durante a visualização. (Maletic et al., 2002). Enquanto que a técnica RELATE consiste em exibir as relações entre os elementos. A *relate* é um dos aspectos mais importantes em softwares de visualização, principalmente quando há hierarquias, na qual há necessidade de enxergar relações entre pai e filho. (Maletic et al., 2002). A técnica SELECT possibilita o usuário selecionar um subconjunto de entidades, parte dos dados, ou até conjuntos inteiros que sejam de seu interesse, afim de obter informações detalhadas ou ocultas dessa seleção. (Khan, M. et al., 2011).

E finalmente, as técnicas de *linking* e *brushing* são utilizadas para vincular elementos e relacioná-los. Os dados podem ser mapeados de uma forma diferente para demonstrar outros pontos de vista ou diferentes frações das informações (Khan, M. et al., 2011).

3.10 ESTUDO DE REDES E TEORIA DE GRAFOS

Historicamente, o estudo de redes está associado à um ramo da matemática discreta denominada teoria de grafos, sendo esta a estrutura natural para o tratamento matemático exato de redes complexas que formalmente podem ser representadas como grafos. (BOCCALETTI, S. et al., 2006)

A teoria de grafos surgiu em 1736, quando Leonhard Euler, um matemático suíço, publicou a solução para o problema da ponte de Königsberg. O desafio era encontrar uma viagem de ida e volta que cruzasse cada uma das pontes da cidade de Königsberg exatamente uma vez. (BOCCALETTI, S. et al., 2006)

Outras inúmeras situações do mundo real podem ser também descritas por um diagrama formado por um conjunto de pontos e de linhas que unem certos pares desses pontos. Como por exemplo, os pontos poderiam representar pessoas, e as linhas unem pares de amigos, ou os pontos poderiam ser centros de comunicação e as linhas representariam as ligações entre estes centros. Nota-se que o principal ponto de interesse desses diagramas dá-se pela informação, se dois pontos estão ligados por uma linha ou não. A abstração

matemática dessas ligações constitui o conceito de grafo (BONDY, J. & MURTY, U., 1976).

Os grafos são assim chamados pela capacidade de serem representados graficamente, e essa representação gráfica auxilia a entender muito de suas propriedades. (BONDY, J. & MURTY, U., 1976)

Um grafo é constituído por um par de conjuntos (V, E) , no qual V é o conjunto de vértices e E é o conjunto de arestas, que tem a função de conectar um ou mais vértices. (RUOHONEN, K., 2013). Cada vértice (V) possui um grau, denominado “*vertex degree*”, que é dado pelo número de arestas incidentes em V . (BONDY, J. & MURTY, U., 1976).

Keijo Ruohonen (2013) descreve alguns dos conceitos fundamentais relacionados a grafos, mesmo afirmando que existem algumas variações destas terminologias. Por exemplo *Caminhadas*: Uma caminhada em um grafo consiste em alternar entre uma sequência de vértices e arestas. (RUOHONEN, K., 2013). O comprimento da caminhada: É o número de arestas na sequência de uma caminhada. (BOCCALETTI, S. ET AL., 2006). Da mesma forma *Caminho*: Um caminho é uma sequência de vértices ligados entre si, em que nenhum nó é visitado mais de uma vez. (Ruohonen, K., 2013; Boccaletti, S. et al., 2006). Enquanto que *Trilhas* é considerado quando em uma caminhada os vértices são visitados no máximo uma vez. (RUOHONEN, K., 2013).

Além disso, define-se como *Circuito*: uma caminhada fechada de, pelo menos, três nós, em que nenhuma aresta é repetida. (BOCCALETTI, S. et al., 2006) e o que *Caminho mínimo* e a caminhada de caminho mais curto de um vértice a outro é definido como caminho mínimo. (BOCCALETTI, S. et al., 2006).

Um outro conceito fundamental é o de grafo não direcionado. Em um grafo não direcionado, cada uma das arestas é definida por um par de nós i e j (Figura 12 a),. A aresta que une os dois nós é dita incidente em i e j . Quando dois nós estão ligados por uma aresta, estes são referidos como adjacentes ou vizinhos. (BOCCALETTI, S. et al., 2006). Enquanto que em um grafo direcionado (Figura 12 b), a ordem do par de nós (i, j) ligados por uma aresta é um fator importante, pois representa uma ligação de i para j . (BOCCALETTI, S. et al., 2006)

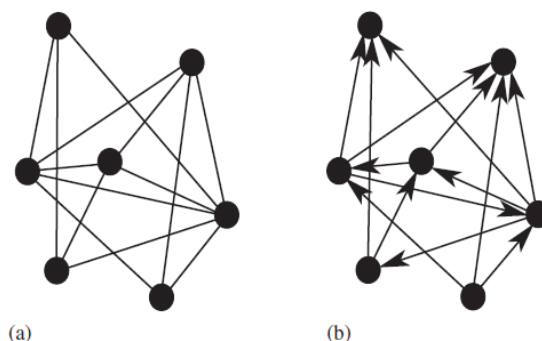


Figura 12 – Representação gráfica de um grafo não direcionado (a) e direcionado (b).

Fonte: Physics Reports 424, p. 180, 2006, adaptado.

3.11 TEORIA DAS REDES COMPLEXAS

A teoria de redes complexas é um campo multidisciplinar que tem conquistado espaço nas mais diversas áreas como a física, engenharia, biologia, sociologia e economia. Muitos dos sistemas encontrados nestes campos científicos possuem uma característica comum: podem ser representados na forma de grafo. (Cherifi, H. 2014). É importante ressaltar que nem todo o grafo é uma rede complexa pois, para isso, deve seguir algumas das específicas propriedades definidas para redes complexas (METZ, J. et al., 2007).

Muitos conceitos sobre redes complexas foram propostos e investigados nos últimos anos. Entretanto, três deles têm sido altamente ressaltados no contexto contemporâneo (ALBERT, R. & BARABÁSI, 2002).

O conceito de *small world* (mundo pequeno, em português), descreve que, mesmo com os tamanhos das redes sendo muitas vezes demasiadamente grandes, na maioria delas existe um caminho curto entre dois nós (Figura 13). Sendo a distância entre dois nós medida pelo número de arestas ao longo do caminho mínimo que os conecta. (ALBERT, R. & BARABÁSI, 2002)

Outras características importantes das redes de mundo pequeno são a vizinhança local é preservada (AMARAL, L. A. et al., 2000) e o diâmetro da rede, que é quantificado pela distância média mais curta entre dois vértices, aumentada logarithmicamente com o número de vértices (AMARAL, L. A. et al., 2000).

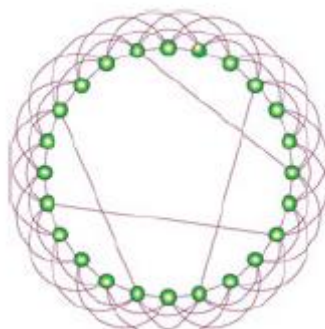


Figura 13 – Rede *Small-world* de Strogatz.
Fonte: Collective dynamics of small-world networks, 1998.

3.11.1 COEFICIENTE DE CLUSTERIZAÇÃO E DISTRIBUIÇÃO DE GRAUS

Nas redes sociais, comumente encontra-se a formação de “*cliques*”, que são grupos exclusivos que podem representar, por exemplo, um círculo de amigos ou conhecidos em que cada membro conhece todos os outros membros. Essa tendência inerente aos nós formarem agrupamento é quantificada pelo coeficiente de clusterização (WATTS & STROGATZ, 1998).

Watts e Strogatz (1998) após várias medições em redes do mundo real, como redes sociais e físicas, afirmam que o coeficiente de clusterização C é igual à probabilidade de que dois vértices vizinhos que compartilham um terceiro vértice sejam vizinhos um do outro. Utilizando a seguinte equação matemática.

$$C = \frac{3 \times (\text{Número de Triângulos do Grafo})}{\text{Número de Triplos conectados aos vértices}}$$

Na equação, “triângulo” refere-se a um trio de vértices em que cada um dos vértices está ligado aos outros. E “triplo conectados” faz referência aos vértices que possuem duas arestas conectando a outros dois vértices. (NEWMAN, M. E. J., 2001). O Fator “3” no numerador é porque cada triângulo possui três triplas e também para assegurar que o coeficiente de clusterização seja um valor entre zero e um (METZ, J. et al., 2007).

Nem todos os nós em uma rede tem o mesmo grau. A distribuição de graus consiste na função de distribuição $P(k)$, que concede a probabilidade de um nó selecionado aleatoriamente tenha k arestas (ALBERT, R. & BARABÁSI, 2002). Os nós de redes aleatórias tendem a possuir o mesmo grau de distribuição e seguem a distribuição de Siméon-Denis Poisson (1838). No entanto, atualmente observa-se que muitas redes se desviam significativamente da distribuição de Poisson, seguindo então a lei de potência (ALBERT, R. & BARABÁSI, 2002; METZ, J. et al., 2007).

3.11.2 MODELOS DE REDES

Os modelos de rede são imprescindíveis para moldar a compressão de redes complexas, bem como a origem das características de uma rede observada. (Barabási & Oltvai, 2004)

O modelo proposto por Erdős e Rényi, denominado ER é o mais simples que uma rede complexa pode assumir. As arestas não-direcionadas são aleatoriamente geradas para um número fixo de nós (METZ, J et. Al, 2007).

Os graus dos nós seguem a distribuição de Poisson, o que indica que todos os nós tem aproximadamente o mesmo “*vertex degree*”. O coeficiente de clusterização é independente do grau de um nó. O comprimento médio do caminho é proporcional ao logaritmo do tamanho da rede, o que significa que esta rede possui a propriedade de *small-world*. (Barabási & Oltvai, 2004)

As redes “*scale-free*” (em português, de livre escala) não seguem a distribuição de Poisson, sendo caracterizadas pela Lei de potência, que consiste da seguinte maneira. A probabilidade de que um nó tenha k ligações segue $P(k) \sim k^{-\gamma}$, onde γ é o expoente do grau. (Barabási & Oltvai, 2004). Portanto, a probabilidade de que um nó esteja altamente conectado é estatisticamente muito mais significativa do que em uma rede aleatória. Essa tendência é conhecida como conexão preferencial, que forma poucos nós, mas com muitas conexões, os denominados *hubs*, e muitos nós com poucas conexões (Barabási & Albert, R., 1999).

Redes hierárquicas são reconhecidas por estar associadas a modularidade. Assume-se que quando módulos combinam de forma hierárquica as propriedades de livre escala com o alto grau de clusterização é gerado o que chama-se por rede hierárquica. (BARABÁSI & BONABEAU, 2003)

O modelo de rede hierárquica integra perfeitamente uma topologia de livre escala com uma estrutura modular. Há a formação de áreas esparsas agrupadas com nós altamente conectados entre si, formando módulos que, com a presença de *hubs*, possuem ligação com módulos vizinhos. (Ver figura 15, parte C). (BARABÁSI & BONABEAU, 2003)

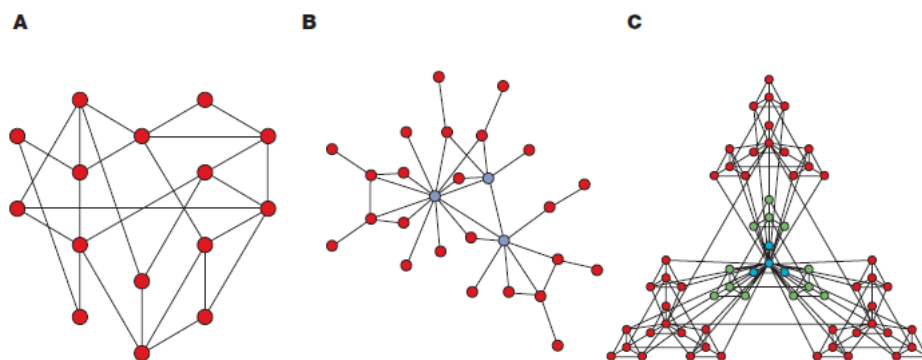


Figura 14 – Exemplo de rede aleatória (A), uma rede “scale-free” (B) e hierárquica (C).
Fonte: Network Biology: Understanding the cell's functional organization, 2004.

3.12 REDES BIOLÓGICAS

A pesquisa biológica utilizando o reducionismo forneceu uma imensidão de informações sobre os componentes celulares individuais e suas devidas funções. Entretanto, nota-se que uma função biológica dificilmente está atribuída a apenas uma única molécula, pois a maioria das características biológicas são originadas de interações complexas entre os componentes celulares como proteínas, DNA, RNA e pequenas moléculas. (Barabási & Oltvai, 2004)

Estudar as células vivas a partir de análise de rede tem trazido perspectivas atraentes sobre o entendimento da estrutura e dinâmica de interações proteína-proteína, vias enzimáticas, vias de sinalização e vias de regulação. As associações desses sistemas geralmente são complexas, possuindo diversos níveis de hierarquias dentro e entre as estruturas da rede. Portanto, a caracterização das propriedades estruturais, análise e visualização de redes têm sido centrais para o estudo e entendimento da biologia de sistemas (BRAY, D., 2003; CASTRO et al., 2012; BARABÁSI & OLTVAI, 2004; LATORA, V. & MACHIORI, M., 2001).

A topologia da maioria das redes biológicas não é completamente regular, nem aleatória, se enquadrando no modelo de rede “scale-free” (WATTS & STROGATZ, 1998).

As redes celulares são “scale-free”, tendo a primeira evidência notada na análise de redes metabólicas, de 43 microrganismos diferentes nos três domínios da vida (*eukaryotes*, *bacteria*, e *archaea*). O estudo indicou que o metabolismo celular possui topologia de livre escala, sendo que a maioria dos substratos participam de apenas uma ou duas reações e possui *hubs* metabólicos com dezenas de ligações (ALBERT, R. et al., 2000; BARABÁSI, 2004).

As redes regulatórias genéticas também fazem parte da arquitetura “scale-free”. Os nós são genes individuais e as arestas são derivadas de dados de correlação de expressão proveniente de *microarray* ou de redes de domínio de proteína (FEATHERSTONE, D. E. & BROADIE, K., 2002).

Em redes biológicas, o conceito de rede direcionada ou não direcionada refere-se a natureza das interações. Redes direcionadas podem representar, por exemplo, reações metabólicas, no qual as arestas representam a direção de um substrato para um produto, ou então a transformação de um fator de transcrição para o gene que o regula. Um exemplo de rede não direcionada,

são as redes proteína-proteína, pois as arestas não tem uma direção atribuída (BARABÁSI & OLTVAI, 2004).

A biologia é repleta de exemplos de modularidade. A maioria das moléculas de uma célula pertence a um complexo intracelular com atividade modular, como o ribossomo, ou atua em um módulo de um processo de amplificação de sinal em uma via de sinalização. Um módulo em uma rede pode ser definido como uma sub-rede que possui mais arestas internas do que externas (BARABÁSI & OLTVAI, 2004; LUO, F. et al. 2007).

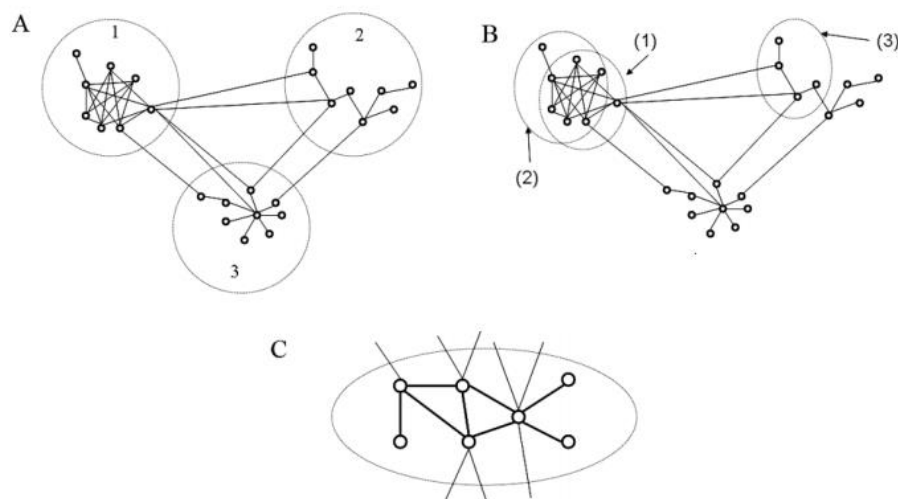


Figura 16 – Em (A), uma rede de amostra incluindo três módulos diferentes(1-3) que são intuitivamente separados por círculos cinza. Em (B), módulos identificados a partir da rede de amostra. (1) Módulo altamente conectado, (2) módulo forte (Radicchi et al., 2004); e (3) módulo fraco (Radicchi et al., 2004). Em (C) uma sub-rede definida como um módulo forte.

Fonte: Modular organization of protein interaction networks, LUO, F. et al. 2007.

Os módulos são grupos de nós que estão relativamente isolados do resto do sistema. No entanto, em uma rede “*scale-free*”, há *hubs* que estão em contato com uma alta fração de nós, sendo improvável a existência de módulos isolados. A combinação de hubs e módulos definem o conceito de rede hierárquica. (BARABÁSI & OLTVAI, 2004)

Capítulo 2



DESENVOLVIMENTO DA PLATAFORMA

CAPÍTULO 2 : DESENVOLVIMENTO DA PLATAFORMA

1. INTRODUÇÃO

Os grandes desafios para construção de uma plataforma web que permita análise e visualização de redes biológicas estão altamente ligados com problemas de caráter técnico. A maioria dos softwares utilizam visualizações planas, não obtendo proveito da característica hierárquica que as redes biológicas possuem, por isso, as redes com maior densidade, ficam difíceis de serem manipuladas e visualizadas.

2. METODOLOGIA E MODELAGEM

Para o desenvolvimento do projeto de software inicialmente foi empregada a metodologia *Scrum* para realizar o planejamento, execução e gestão da plataforma.

Os componentes principais do *Scrum* são:

- *Product Backlog*: Lista com os requisitos do projeto (CERVONE, H. F., 2011).
- *Product Owner*: Responsável por gerenciar o projeto. Pode alterar as prioridades e características, aceita ou rejeita os resultados do trabalho e define o *product backlog* (SUTHERLAND, J., 2014).
- *Scrum Master*: Protege a equipe de interferência externas, garante que a equipe permaneça totalmente funcional e produtiva (SUTHERLAND, J., 2007).
- *Scrum Team*: Equipe multifuncional, auto-organizada que trabalha no projeto em tempo integral (CERVONE, H. F., 2011).

O uso do *Scrum* tem cinco atividades principais: o *Kickoff*, a reunião de planejamento de uma *Sprint*, a *Sprint*, o *Daily Scrum*, e a reunião de revisão da *Sprint*. (CERVONE, H. F., 2011)

O planejamento de um *Sprint* ocorre através de uma reunião entre o *Scrum Team*, o *Scrum Master* e o *Product Owner*. Sendo que nesta reunião define-se o *product backlog* e os resultados formais desta *Sprint*. A reunião de *kickoff* é bem semelhante a esta, a diferença é que são estabelecidos um *backlog* de alto nível e os objetivos gerais do projeto (CERVONE, H. F., 2011).

Uma vez que foi feita a reunião de planejamento da *Sprint*, essa pode ser iniciada. As *Sprints* são limitadas a um ciclo de iteração de um mês, período em que a funcionalidade é desenvolvida. Dois fatores importantes que devem ser ressaltados é que durante a execução de uma *Sprint* nenhuma influência externa deve interferir no trabalho e nenhum dos requisitos deve ser alterados. O *Daily Scrum* são reuniões diárias de até 15 minutos para disseminar as atividades realizadas e identificar possíveis dificuldades (CERVONE, H. F., 2011).

MVC

Na criação de aplicações interativas, modularizar os componentes tem inúmeras vantagens, pois proporciona uma maior facilidade no entendimento e na manutenção da aplicação (KRASNER, E. G. & POPE, S. T., 1988). Por isso, para a arquitetura do projeto optou-se em seguir o padrão MVC.

O MVC é um padrão de arquitetura de software que propõe a independência através da dissociação do acesso aos dados, da lógica de processamento de dados e a visualização dos dados. Sendo dividido em três camadas: (KRASNER, E. G. & POPE, S. T., 1988; CURRY, E., 2008)

- *Model*: É a camada que contém a implementação da estrutura central da aplicação.
- *View*: É camada que apresenta dados na tela. Nela ocorrem as interações do usuário, serve para solicitar dados do modelo e exibi-los, assim como mensagem de resposta.
- *Controller*: Os controladores são utilizados para realizar a comunicação das *Views* com o modelo.

UML

Seguindo o paradigma da orientação a objeto com a arquitetura MVC, para uma representação padronizada da arquitetura do sistema, optou-se pelo uso da Unified Modeling Language (UML).

Segundo Medvidovic et al. (2002), pode-se descrever a UML como um conjunto padrão de modelos de design para representar a arquitetura de um software. Os modelos da UML são parciais e abordam diversas questões em determinados níveis de fidelidade usando uma variedade de diagramas. Os diagramas representam:

1. Classes, seus atributos, operações e relacionamentos.
2. Os comportamentos e estados possíveis de classes individuais.
3. Os pacotes de classes e suas devidas dependências.
4. Casos de uso do sistema, incluindo os tipos de usuário e suas respectivas tarefas.
5. O comportamento global do sistema no contexto de um caso de uso.
6. Exemplos de instâncias de objetos com atributos e relações em um determinado cenário.

3. PLANO DE ATIVIDADES

O plano de atividades demonstra como foi planejada a execução do projeto e o fluxo de trabalho do *Scrum Team*, sendo este representado pelos seguintes diagramas: Work Breakdown Structure (WBS) e Gantt.

DIAGRAMA WBS

A WBS (*work breakdown structure*) refere-se ao conjunto de tarefas e marcos que deverão ser atendidos dentro de um projeto. Cada tarefa representa a criação ou modificação de uma parte do produto (SERVELLO, M., 2002).

Na figura 17, o diagrama WBS está dividido em 5 tarefas principais a serem realizadas. Estas tarefas possuem vários níveis de hierarquias, que definem maior grau de especificidade.

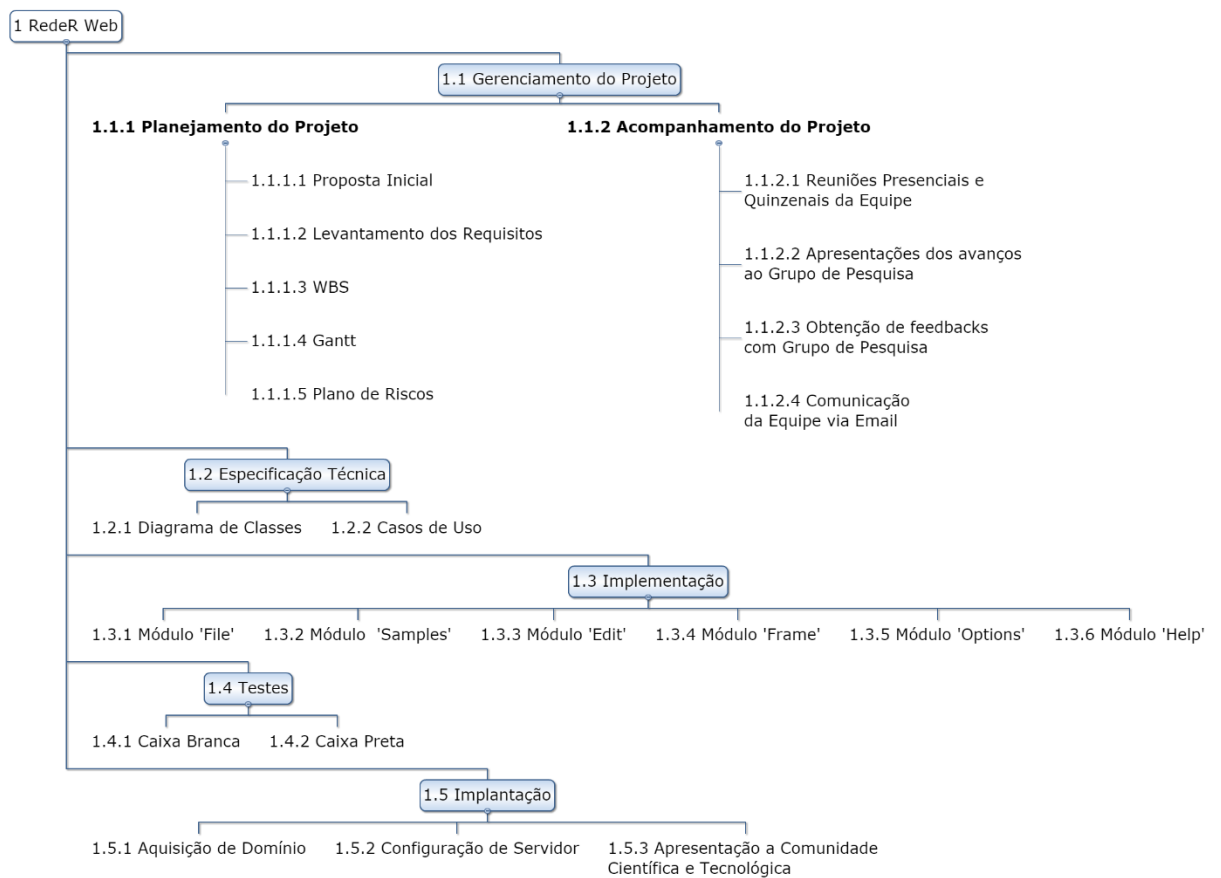


Figura 17 – Diagrama WBS para desenvolvimento do RedeR Web.

DIAGRAMA GANTT

O gráfico de *Gantt* é um instrumento vantajoso para a apresentação de tempo relativo a um planejamento de um projeto. Podendo representar graficamente previsões ou realizações passadas. (MAYLOR, H., 2001).

Para demonstrar o cronograma de execução da plataforma, a figura 18 representa o gráfico de *Gantt* para as principais etapas estabelecidas na WBS.

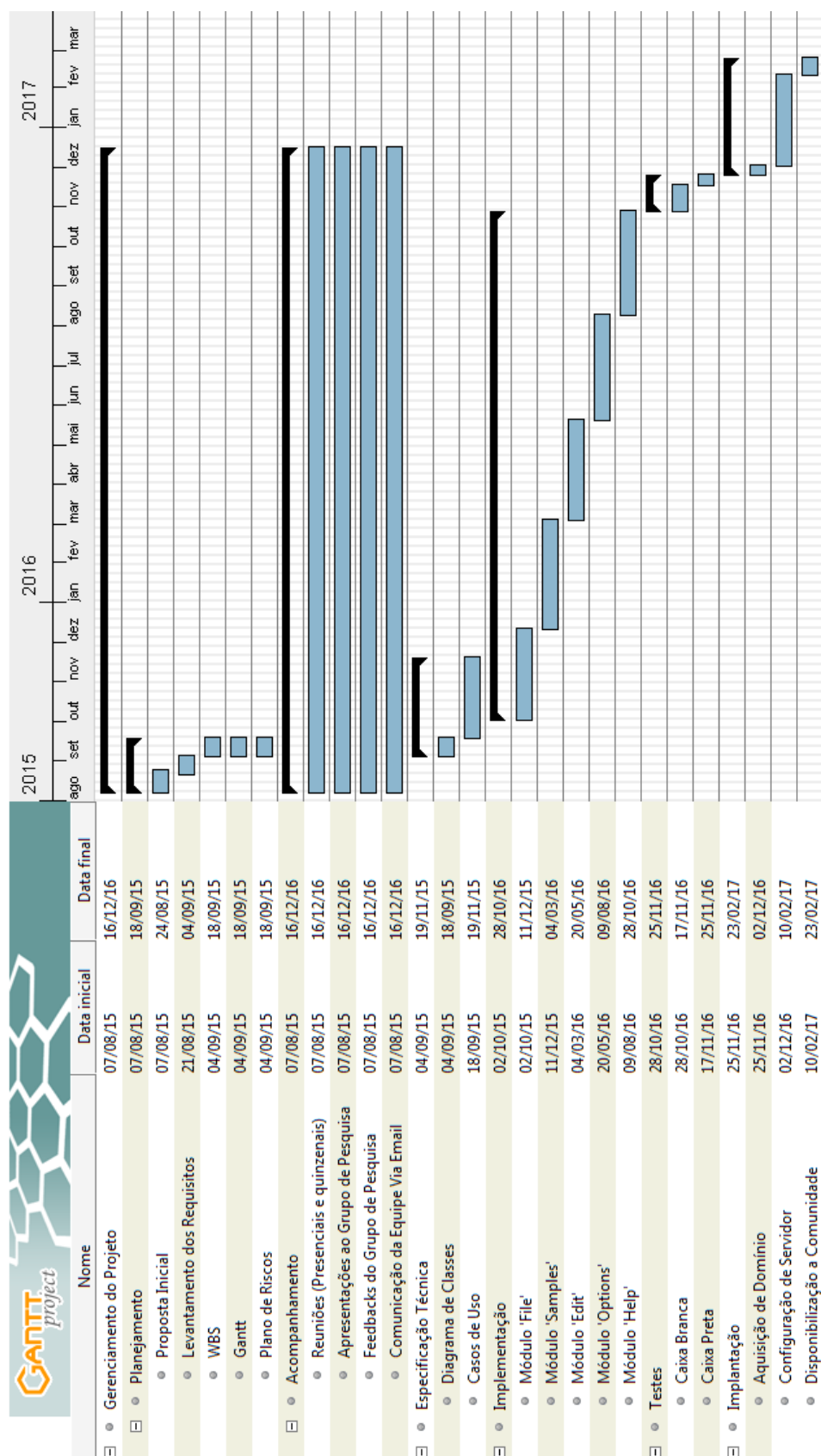


Figura 18 – Diagrama Gantt para desenvolvimento do RedeR Web.

4. PLANO DE RISCOS

A ocorrência de um evento negativo que afete o desenvolvimento de um projeto é considerado um risco. Para antecipar e minimizar os efeitos de eventos que possam impactar o projeto, deve-se realizar o gerenciamento de riscos. Para isso, pode-se seguir as atividades a seguir: a identificação, análise, planejamento, acompanhamento e resolução dos riscos. (Aguilar, M., 2003).

Visando estabelecer, controlar e garantir a execução bem sucedida do projeto, estabeleceu-se as seguintes métricas para montar o plano de riscos representado pela tabela 1:

1. ID: Identificação do Risco
2. Fator de risco: Refere-se a descrição do risco.
3. Consequência: Efeito negativo na execução do projeto.
4. Ação: Atitude tomada para a resolução do risco.
5. Probabilidade: Métrica utilizada pra representar a probabilidade de um risco ocorrer. As probabilidades são caracterizadas da seguinte maneira:
 - Muito baixo: 20% de probabilidade.
 - Baixo: 20% a 40% de probabilidade.
 - Moderado: 40% a 60% de probabilidade.
 - Alto: 60% a 80% de probabilidade.
 - Muito Alto: 80% a 100% de probabilidade.
6. Importância:
 - Baixa: Sem redução da qualidade do software, ocasionando em atrasos na entrega de até 5%.
 - Média: Com possibilidades mínimas de redução de qualidade, ocasionando atrasos de 5% a 10%.
 - Alta: Redução da qualidade das principais funcionalidades, com atrasos que podem ir de 15% a 30%.
 - Muito alta: Riscos que podem impossibilitar a entrega do software.

ID	Fator de Risco	Consequência	Ação	Probabilidade	Importância
1	Problemas na definição dos requisitos	Falhas na documentação seguido de falhas na implementação	Reuniões específicas para definições e validações dos requisitos	Baixa	Alta
2	Incompatibilidade no uso de bibliotecas: D3.JS, JQuery Mouse Wheel, JQuery UI.	Atraso na implementação	Caso não haja solução na implementação, buscar por bibliotecas similares para substituição	Moderada	Alta
3	Problema em otimização da renderização	Lentidão na interatividade do software	Investigar motivos, e se necessário trocar bibliotecas que renderizam os elementos	Baixa	Muito Alta
4	Novas necessidades encontradas	Replanejamento das atividades, possível perda de qualidade do software	Adequação do escopo inicial	Moderada	Alta
5	Problemas com a infraestrutura de desenvolvimento	Atraso no desenvolvimento dos módulos	Busca por novo equipamento, readequar cronograma	Baixa	Baixa

Tabela 1 – Plano de Riscos

5. TECNOLOGIAS

SUBVERSION

Para controle de versão da plataforma foi utilizado o Apache Subversion, que é um projeto *open source* fundado em 2000 pela CollabNet, Inc.

Com o Subversion é possível gerenciar todo o código-fonte da aplicação. Ele gerencia todas as alterações feitas em pastas e arquivos, permitindo que seja mantido um histórico de versões do software, podendo realizar comparativos entre as versões, retroceder se alguma alteração foi indevida (PILATO, M. C. et al., 2008).

O Subversion pode operar em servidores, o que significa que várias pessoas podem cooperar no desenvolvimento de uma aplicação atuando no mesmo conjunto de arquivos (PILATO, M. C. et al., 2008).

JAVA EE

A tecnologia JAVA é uma linguagem de programação e uma plataforma. A linguagem é de alto nível e segue o paradigma de orientação a objetos, possuindo sintaxe e estilos particulares. A plataforma Java é um ambiente específico nos quais as aplicações da linguagem de programação são executadas (Oracle, 2012).

O JAVA SE (Standart Edition) define as principais funcionalidades Java como os objetos básicos, as classes utilizadas para rede, segurança, acesso a banco de dados, análise de arquivos XML. (Oracle, 2012).

O JAVA EE (Enterprise Edition) foi construído sobre o Java SE, para que passasse a fornecer a possibilidade do desenvolvimento e execução de aplicações Web em grande escala, com multicamadas escaláveis, confiáveis e seguras. (Oracle, 2012)

Visando compatibilizar ao máximo com a versão *stand-alone* do RedeR, desenvolvido em Java SE. A linguagem de programação utilizada foi a Java EE, permitindo a utilização de vários pacotes e classes já empregadas no RedeR, como o pacote *Model* e as classes de manipulação de arquivos.

ECLIPSE NEON

Para desenvolver o software utilizou-se o Eclipse Neon (<http://www.eclipse.org>), que é uma IDE para desenvolvimento JAVA. Ela possui uma série de ferramentas e plug-ins que auxiliam na produção de aplicações Web.

GLASSFISH SERVER 4.0

Para desenvolver e implantar o RedeR Web utilizou-se o GlassFish Server 4.0 (<http://glassfish.java.net/>) que é um servidor *open source* capaz de fornecer recursos para desenvolvimento de aplicações Java EE.

O Glassfish Server 4.0 proporciona principalmente estabilidade, compatibilidade e segurança para os projetos implantados. (Oracle, 2013)

JQUERY 1.11.0

O JQuery (<https://jquery.com/>) é uma biblioteca de JavaScript rápida, leve e rica em recursos, simplificando a manipulação de elementos HTML, controle de eventos, animações e requisições assíncronas. Sendo capaz de funcionar em praticamente todos os navegadores.

JQUERY MOUSE WHELL 3.1.13

O JQuery Mouse Whell (<https://github.com/jquery/jquery-mousewheel>) é um plug-in do JQuery que fornece suporte para controlar a roda do mouse. Neste projeto utilizou-se para alternar entre os eventos do mouse como *zoom* e *brush*.

JQUERY SPECTRUM 1.8.0

O JQuery Spectrum (<http://bgrins.github.io/spectrum/>) é uma biblioteca do JQuery que fornece um espectro de cores. Foi utilizado para permitir que o usuário selecione as cores do background, do nó, da aresta, contornos entre outros.

JQUERY UI 1.11.4

O JQuery UI (<https://jqueryui.com/>) é uma biblioteca de JavaScript que contém um conjunto de interações do usuário, efeitos, temas e *widgets*. Foi utilizado para gerenciar as janelas de diálogo, na validação e criação de formulários.

D3 3.0

A D3.js (<https://d3js.org/>) é uma biblioteca JavaScript que serve para manipular documentos com base em dados, produzindo visualizações interativas com o uso de HTML, SVG e CSS.

A D3 fornece, além de extrema rapidez, o suporte a grandes volumes de dados permitindo que possuam comportamento dinâmicos com interações e animações. Além disso, possui também uma coleção de componentes e plug-ins.

No desenvolvimento do projeto, a D3 foi a biblioteca fundamental, pois foi utilizada para carregar e renderizar os elementos, aplicar as técnicas de *zoom*, *pan*, *brush* e outros controles de visão.

6. INFRAESTRUTURA

As máquinas utilizadas para desenvolvimento e testes foram:

1. Sony Vaio VPCE40EB/W – Intel Core I3 com 6Gb de memória ram e 250Gb de disco.
2. Dell Optiplex 7010 – Intel Core I5, 8Gb de memória ram e 500Gb de disco.

7. FORMATOS DE ARQUIVOS

XGMML

O formato XGMML (eXtensible Graph Markup and Modeling Language) é utilizado para descrições de grafos. O XGMML utiliza tags para descrever nós e arestas. Sendo uma forma padronizada, possibilita a troca de informações entre diferentes ferramentas de criação e visualização de grafos. Alguns softwares como Cytoscape, Biomax e o RedeR utilizam este formato.

```
<node label="n1" id="0">
  <att type="string" name="alias" value="n1" />
  <att type="float" name="node.bend" value="50.0" />
  <att type="float" name="node.size" value="20.0" />
  <att type="hexadecimal" name="node.labelColor" value="#333333" />
  <att type="string" name="node.font" value="Arial-0-12" />
  <att type="double" name="node.weight" value="0.0" />
  <att type="float" name="font.x.position" value="10.0" />
  <att type="float" name="font.y.position" value="12.0" />
  <att type="string" name="node.shape" value="ellipse" />
  <graphics type="ELLIPSE"
    h="20.0" w="20.0" x="197.49356" y="194.86026"
    fill="#ffcccc" width="1.0" outline="#9999ff"
  />
</node>
```

Figura 19 - Representação de um nó e seus atributos no formato XGMML.

```

<edge label="8" source="1" target="9" direction="0">
  <att type="float" name="arrowlength" value="5.0" />
  <att type="float" name="arrowangle" value="20.0" />
  <graphics width="1.0" fill="#9999ff" edgeLineType="SOLID" />
</edge>

```

Figura 20 - Representação de uma aresta e seus atributos no formato XGMML.

Um dos formatos que RedeR Web trabalha é o XGMML, podendo ler e gravar as redes criadas ou visualizadas.

JSON

O formato JSON (JavaScript Object Notation) é um formato de texto para a serialização de dados estruturados sendo uma derivação dos objetos do JavaScript. Os principais objetivos do formato JSON é ser mínimo, portátil, textual e adequado ao JavaScript. Podendo representar quatro tipos primitivos (Strings, Números, Booleanos e nulos) e dois tipos estruturados. (Objetos e Arrays). (Crockford, D., 2006).

Como a biblioteca D3 fornece a possibilidade de carregar um objeto JSON contendo nós e arestas, após o RedeR Web carregar um arquivo XGMML estes dados são encapsulados nos objetos do modelo e depois convertidos para o formato JSON.

```

{
  "source": 1,
  "target": 18,
  "timeVector": {},
  "edgeColor": {
    "value": -6710785,
    "falpha": 0.0
  },
  "lineWidth": 1.0,
  "shapeType": "SOLID",
  "corDestaque": {
    "value": -13312,
    "falpha": 0.0
  },
  "destaque": false,
  "edgeWeight": 0.0,
  "normalizedWeight": 0.0,
  "normOutEdgeScale": 1.0,
  "isToHide": false,
  "defaultEdgeAssignment": true,
  "isOutEdge": false,
  "isOldEdge": true,
  "arrow": 0,
  "units": 1,
  "ARROW_RELATIVE_LENGTH": 5.0,
  "OPENING_ANGLE": 20.0,
  "edgeID": 17
}

```

Figura 21 – Exemplo de uma aresta e seus atributos estruturados no formato JSON.

8. RESULTADOS E APRESENTAÇÃO DA PLATAFORMA

A plataforma desenvolvida, intitulada RedeR Web, está sendo submetida ao INPI (Instituto Nacional da Propriedade Industrial) para registro de software e será disponibilizada no domínio: <http://www.rederweb.com.br>, já adquirido no Registro BR (<https://registro.br/>).

Diferentemente do Cytoscape.js e outras bibliotecas que visam fornecer visualização e manipulação de redes na WEB, o RedeR Web é uma aplicação Web completa, não necessitando ao usuário ter conhecimento de programação para utilizá-la.

O RedeR Web fornece uma interface intuitiva e responsiva, para que a experiência do usuário com a plataforma possa ser facilmente estimulada e o usuário preocupe-se apenas em realizar sua tarefa principal, sem ter dificuldades no uso.

Na primeira vez que o usuário acessa o softwares, é exibida uma tela de apresentação ressaltando as principais características como a modularidade, a análise e visualização em diferentes níveis hierárquicos, o formatos de entrada e saída disponíveis e outras informações complementares. Em seguida o usuário pode clicar em “Start” e iniciar a navegação pela plataforma.

O RedeR Web torna-se uma ferramenta poderosa trabalhando de forma eficaz com redes de até 50.000 elementos, a partir deste número a plataforma pode apresentar alguns *delays* variando de acordo com a configuração do computador do usuário.

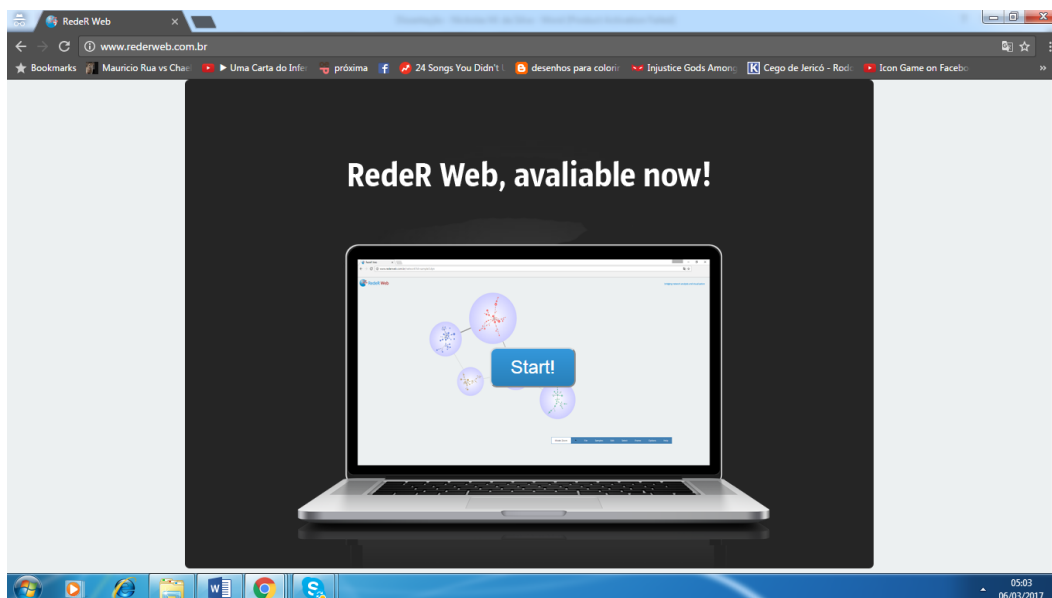


Figura 22 – Splash Screen: Somente na primeira vez que o usuário acessa o RedeR Web é exibida a tela de apresentação.

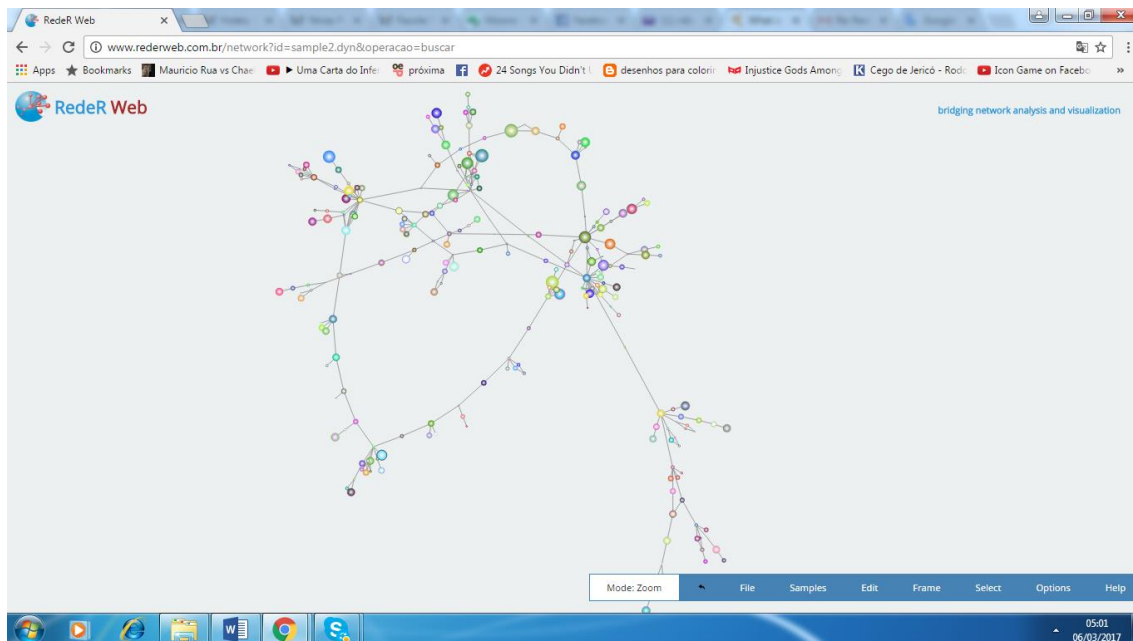


Figura 23 – Tela inicial do RedeR Web com o "Sample 2" carregado, sendo este acessado pelo menu "Samples" -> "Sample2".

PRINCIPAIS FUNCIONALIDADES

IMPORTAÇÃO E EXPORTAÇÃO DE DADOS EM DIVERSOS FORMATOS

A vantagem de uma plataforma permitir a importação e exportação de diferentes formatos de arquivos é a compatibilidade com outros sistemas, proporcionando ao usuário maior flexibilidade. Acessando o menu "File" pode-se importar ou exportar:

- Uma rede completa no formato XGMML contendo todos os nós, arestas e atributos como cores, *shapes*, posicionamentos, relacionamentos
- Nós ou arestas no formato TXT e DAT.
- Uma "Basic Network", contendo os nós e seus atributos básicos como id, *alias*, *shape*, tamanho, cor e posicionamento.
- Arquivo JSON contendo todos os nós, arestas e seus atributos.

Além destas opções, é possível exportar imagens em alta qualidade no formato Portable Network Graphics (PNG) e Portable Document Format (PDF).

SELEÇÃO DE NÓS E ARESTAS

O método de seleção pode ser acessado pelo menu "Select". Quando o usuário posiciona o mouse sobre esta opção, ele pode optar por selecionar toda a rede, anular alguma seleção ou filtrar de acordo com algum parâmetro definido pelo usuário. Após o usuário confirmar a operação, a plataforma deixará em destaque os nós resultantes da pesquisa.

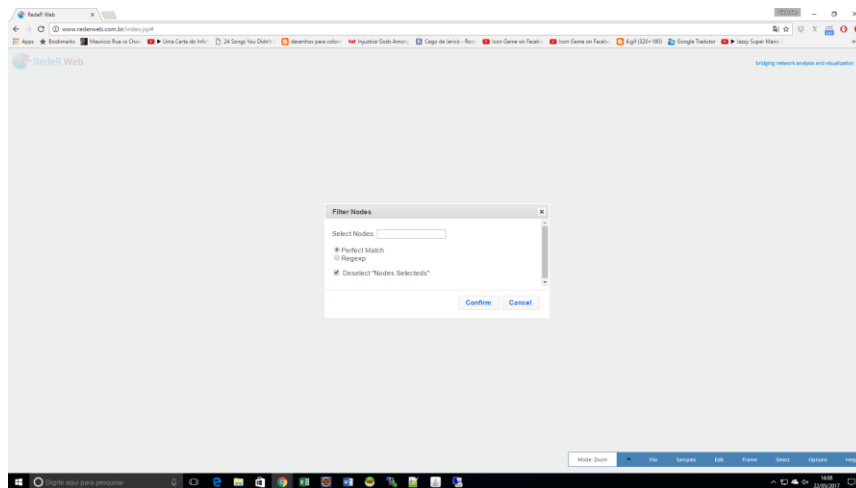


Figura 24 – Demonstração da Funcionalidade de Seleção. O usuário pode optar se a busca será por uma combinação perfeita (perfect match), ou por uma expressão regular (regex).

ZOOM E PAN

Utilizando o mouse o usuário pode facilmente combinar o ZOOM e PAN para ter uma escalabilidade completa da interface. Podendo obter diferentes tipos de visões.

BRUSH

Com a ferramenta de Brush, pode-se selecionar um ou mais elementos dentro da rede clicando e arrastando pela interface. É exibido um retângulo amarelo na área de seleção.

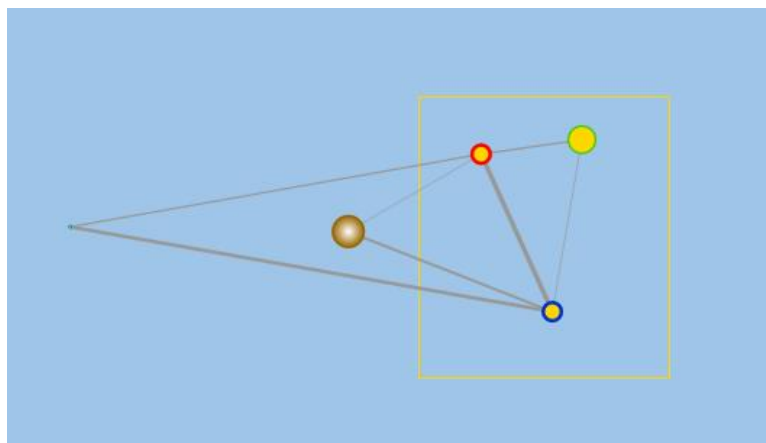


Figura 25 – Demonstração do uso do “Brush”.

ADIÇÃO/EXCLUSÃO DE NÓS E ARESTAS

Para adicionar um novo nó ou aresta, basta clicar com o botão direito do mouse no lugar desejado para a inserção do elemento e clicar em “Add Node”.

Após isso, é solicitado a inserção do atributo identificador “*alias*”. Caso o usuário preencha com algum identificador já existe, será criado uma relação entre os Node A e o Node B.

Para excluir um elemento, basta clicar no elemento com o botão direito e selecionar a opção “delete” no menu de contexto.

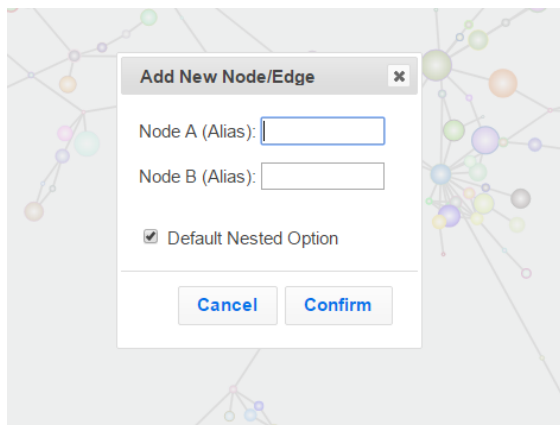


Figura 26 – Demonstração de adição de nó/aresta.

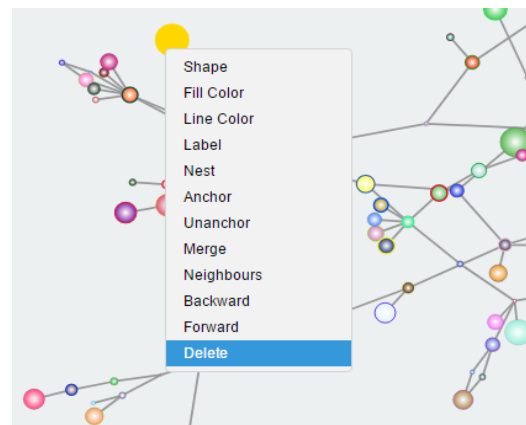


Figura 27 – Demonstração do menu de edição de um nó, selecionando a opção de remoção de nó/aresta.

EDIÇÃO DE ATRIBUTOS DOS NÓS

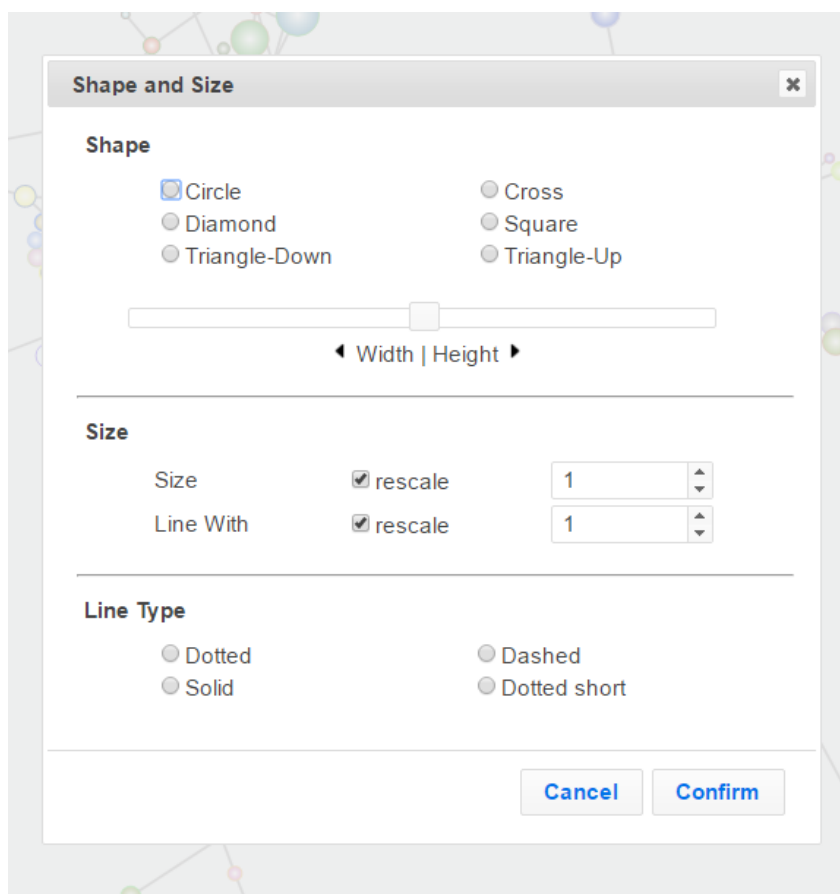


Figura 28 – Demonstração de adição de nó.

Os atributos dos nós podem ser editados para serem visualmente destacados, seja por tamanho, por cor, *shape*. Para editar um ou mais nós, deve-se executar a ferramenta de seleção e clicar no nó desejado com o botão direito. (Ver Figura 27)

EDIÇÃO DE ATRIBUTOS DAS ARESTAS

O mesmo mecanismo para a edição dos nós é utilizado para a edição de arestas. Pode-se editar o tipo da interação, cor, peso.

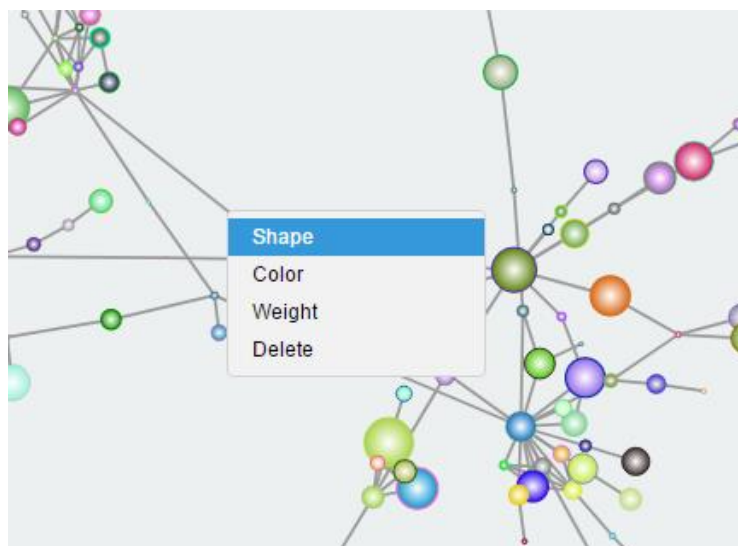


Figura 29 – Menu de edição das arestas

ALTERAÇÃO DE PLANO DE FUNDO

Pode-se trocar o plano de fundo para adequar e destacar determinadas ligações ou nós. Acessando o menu “Frame->Background”, é exibida um espectro de cores permitindo ao usuário que escolha aquela que mais se adeque aos seus dados.

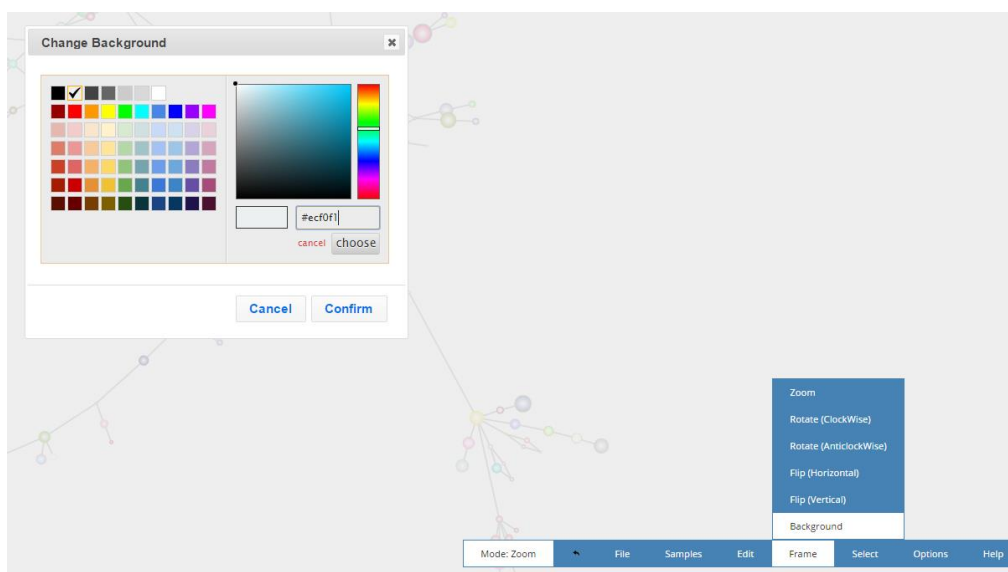


Figura 30 – Menu de edição de Plano de Fundo

FORMAÇÃO DE CONTAINERS/MÓDULOS

Em redes hierárquicas é fundamental que a plataforma exiba estas relações de pai e filho de uma forma simples e intuitiva. Pode-se criar módulos da seguinte maneira:

- Selecionar vários nós e criar um módulo que será o pai para eles.
- Transformar um ou mais nós em módulos que poderão receber outros nós utilizando a função “*Migrate*”.

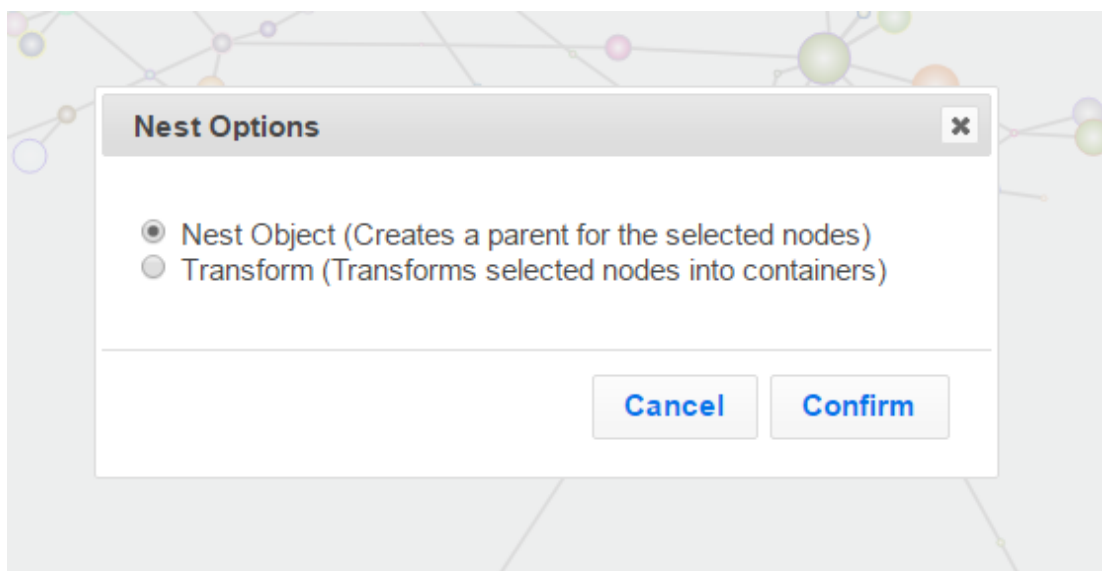


Figura 31 – Clicando com o botão direito do mouse em um nó, pode-se escolher a opção “Nest” permitindo o usuário a optar pelo tipo de criação de módulos.

MIGRAR

A plataforma permite ao usuário que facilmente seja alterado a “paternidade” de um conjunto de nós ou até mesmo de um módulo todo. Para isso basta arrastar a seleção desejada e soltar na área do módulo correspondente (Ver Figura 32).

DESFAZER/REFAZER ALTERAÇÃO

Uma funcionalidade que a comunidade adepta ao RedeR manifestou interesse é a função de desfazer algum dos movimentos. Isto é imprescindível a uma plataforma que está enquadrada na visualização de dados. Permitir que o usuário navegue em sua construção de uma determinada rede, avançando ou retrocedendo etapas.

- Para desfazer uma operação pode-se utilizar o ícone presente no menu principal, ou usar a combinação de teclas CTRL+Z
- Para refazer uma operação, basta combinar as teclas CTRL+Y

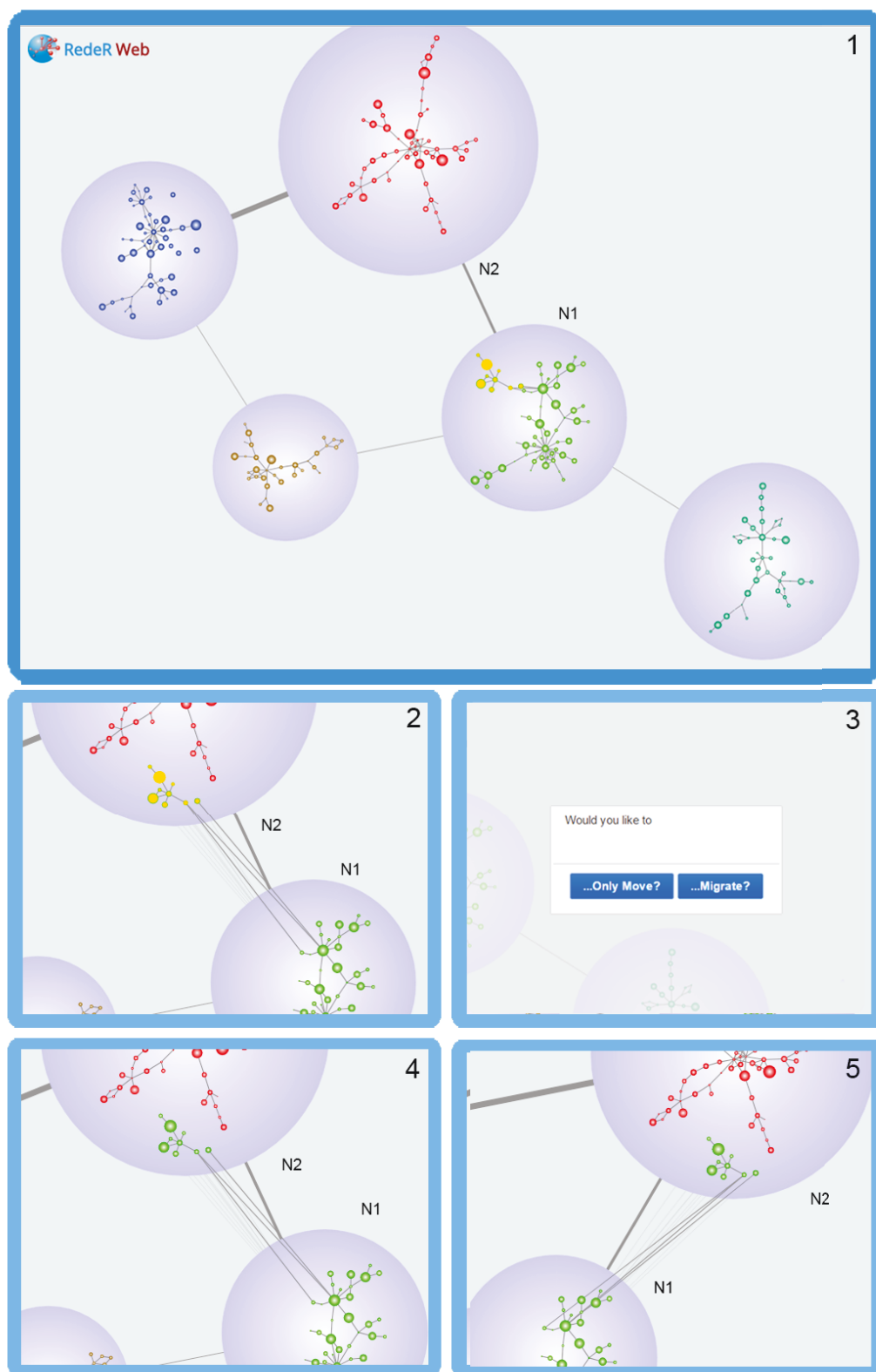


Figura 32 – Demonstração da funcionalidade “Migrate”. 1). É realizado uma seleção. 2). A seleção é movida de N1 para N2. 3). A plataforma ao detectar que o conjunto de nós foi movida para uma região na qual está localizada outro módulo, pergunta ao usuário se quer migrar ou apenas mover. 4). Ao selecionar “migrate”, os nós selecionados deixam de ser filhos de N1 e passam a se relacionar com N2. 5). Ao movimentar N2, pode-se observar que os nós que foram migrados, caminham junto com N2.

MERGE OUT-EDGES

Quando se trata de redes hierárquicas com um grande volume de nós e arestas, a tendência é que a visualização seja mais complicada. A função “Merge-out-edges”, remove as ligações entre nós de um módulo com nós de outro módulo, podendo o peso dessas arestas serem somados a ligação Módulo->Módulo ou recalculados por um algoritmo definido na plataforma. Com isso, o RedeR Web reduz a densidade de arestas de uma rede altamente complexa.

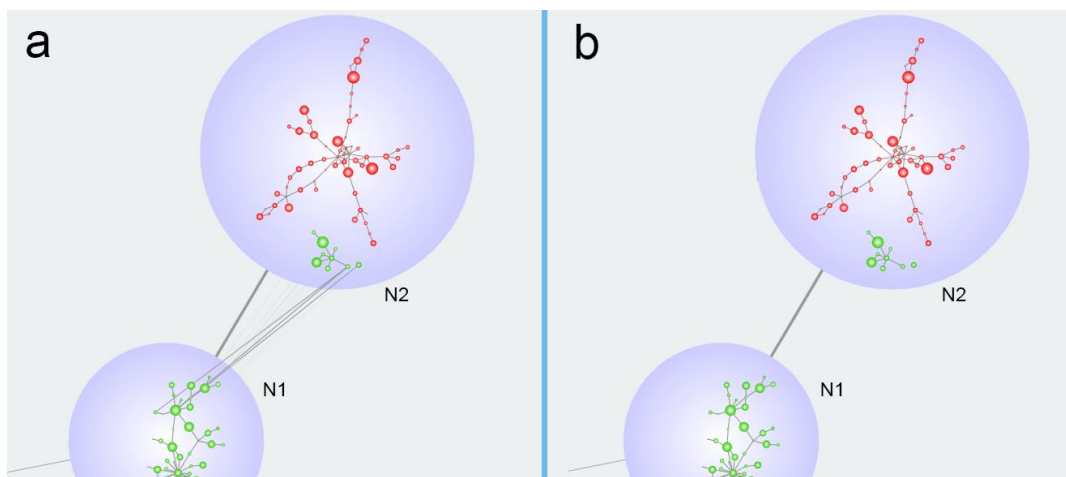


Figura 33 – Função de Merge-Out-Edges. Em **a** é notório a existência de várias ligações entre nós do módulo N2 com nós do N1. Em **b** após a execução da função “Merge-Out-Edges”, as ligações são removidas e somadas a ligação de módulo para módulo.

HOSPEDAR E COMPARTILHAR

A função de “Hosting and Sharing” permite ao usuário compartilhar suas redes com outro usuário. A plataforma gera um link único e criptografado, sendo possível cadastrar uma senha para permitir edição. Esta funcionalidade pode-se ser muito útil para pesquisadores, pois permite que vários usuários trabalhem com a mesma rede.

Hospedar uma rede no RedeR Web também pode ser utilizado para artigos publicados, pois as figuras de redes produzidas pela comunidade podem ser acessadas por um link de protocolo http. Isso faz com que o leitor do artigo possa obter maiores detalhes sobre o estudo em questão.

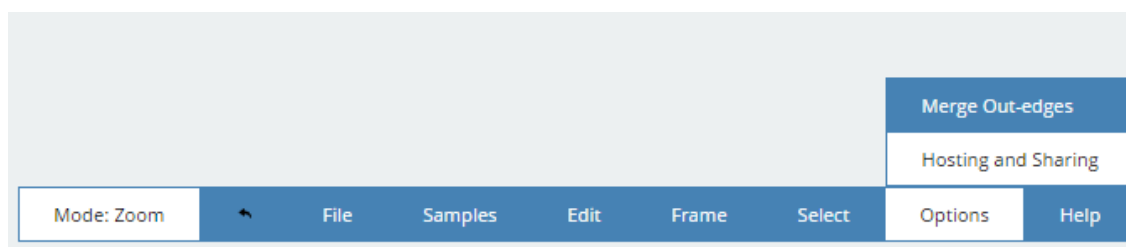


Figura 34 – Função de Hospedar e Compartilhar. Basta clicar no menu “Options”->”Hosting and Sharing”.

ESTRUTURA DE DADOS

A estrutura de dados do RedeR Web foi implementada de maneira igual à do RedeR (Figura 35), na qual há uma modelagem única capaz de armazenar e manipular os dados hierárquicos. Essa estrutura emula um gráfico com duas camadas e diversos níveis.

Pode-se notar que a primeira camada (Figura 2a) há uma representação de uma árvore de dados com múltiplos ramos e níveis e sem ciclos. Apresentando um dos pais para cada vértice e podendo ser definida como um gráfico acíclico direcionado (DAG). A segunda camada é projetada como um gráfico não direcionado no nível mais baixo da hierarquia (DAG). (Castro et al., 2012)

Um fator importante na modularização da estrutura de dados é que as associações de caráter horizontal ou não-hierárquica são reatribuídas de uma camada para outra. Para o usuário, essa estrutura será exibida como na figura 35b do lado direito, podendo representar conjuntos de dados como interações proteína-proteína, redes, ontologias, entre outros. (Castro et al, 2012)

Executando um comparativo entre as redes hierárquicas e planas, conclui-se que a organização modular comporta e exibe a informação completa, já as redes planas (figura 35b do lado esquerdo) não sendo dividida em módulos, podem apenas representar uma instância de linha da estrutura de dados.

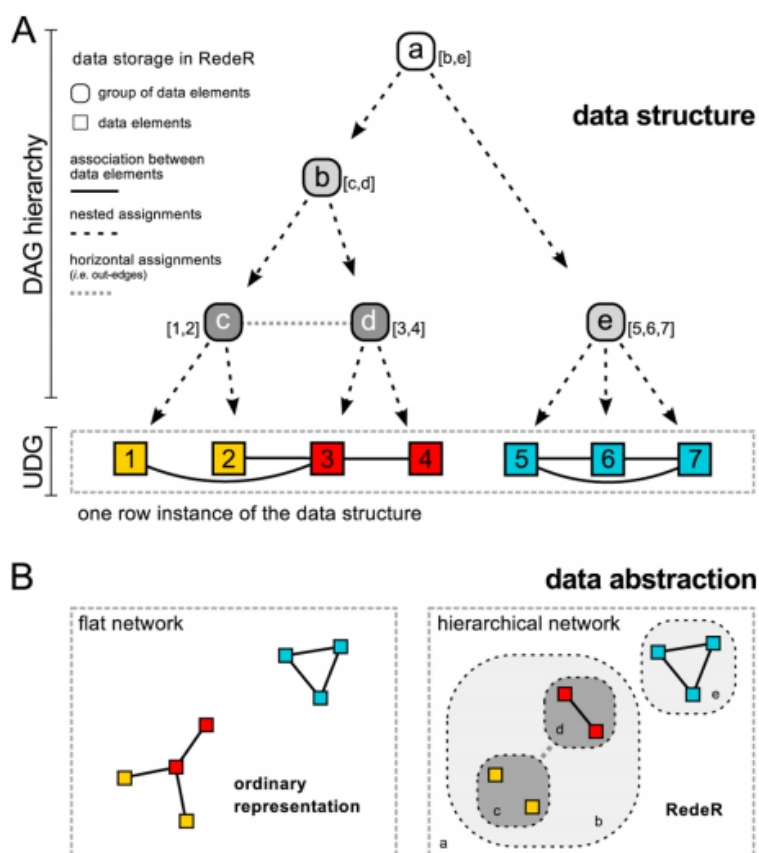


Figura 35 – Como o RedeR organiza e manipula sua estrutura de dados. (Castro et. Al, 2012)

DIAGRAMA MVC

O diagrama MVC (Model-View-Controller) ilustra como o RedeR Web encapsula a estrutura de dados e torna capaz de o usuário visualizar e manipular a informação partindo de um panorama geral até um nível específico sem que se perca alguma informação.

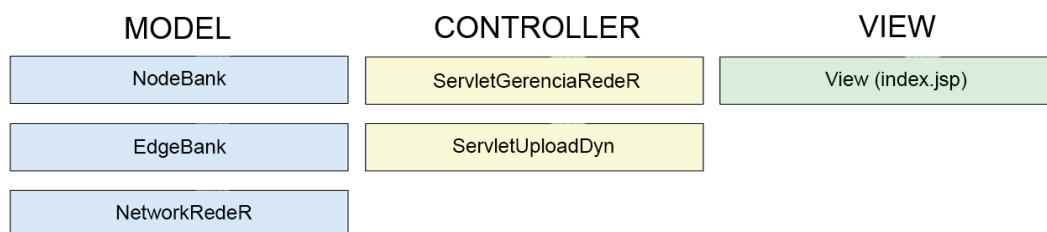


Figura 36 – Diagrama (Model-View-Controller)

DIAGRAMA DE CLASSES

O diagrama de classes (Figura 37) apresenta as estruturas das classes que compõe o modelo assim como seus devidos relacionamentos.

9. TESTES

Os testes foram feitos com uma instância do RedeR Web em um servidor Glassfish 4.1, em uma máquina Dell Optiplex 7010 I5, com 8GB de memória ram. Utilizando os navegadores Google Chrome, Mozilla Firefox e Ópera.

*O número de elementos de cada rede de teste pode ser observado a abaixo:

- 1- 12 elementos. (5 nós e 7 arestas).
- 2- 491 elementos (246 nós e 245 arestas).
- 3- 782 elementos (251 nós e 531 arestas).
- 4- 3273 elementos (936 nós e 2337 arestas).
- 5- 5050 elementos (100 nós e 4950 arestas).
- 6- 20100 elementos (200 nós e 19900 arestas).

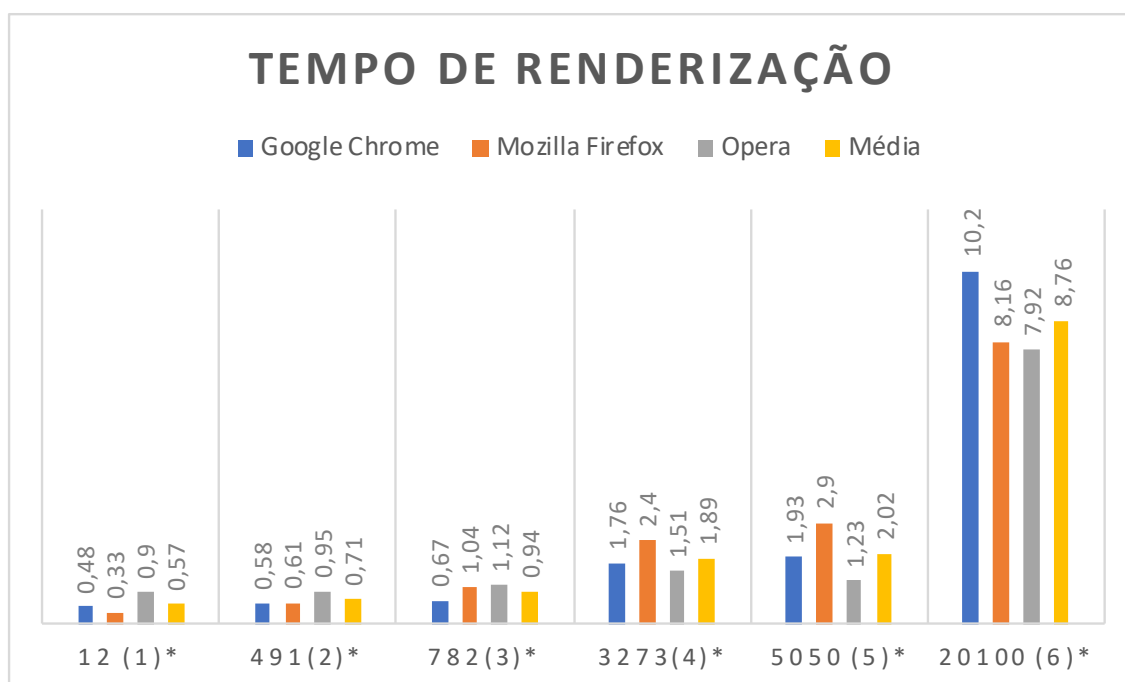


Figura 41 – Relatório de Testes: tempo em segundos para renderizar redes de diferentes tamanhos entre navegadores executando o RedeR Web. Fonte: O Autor.

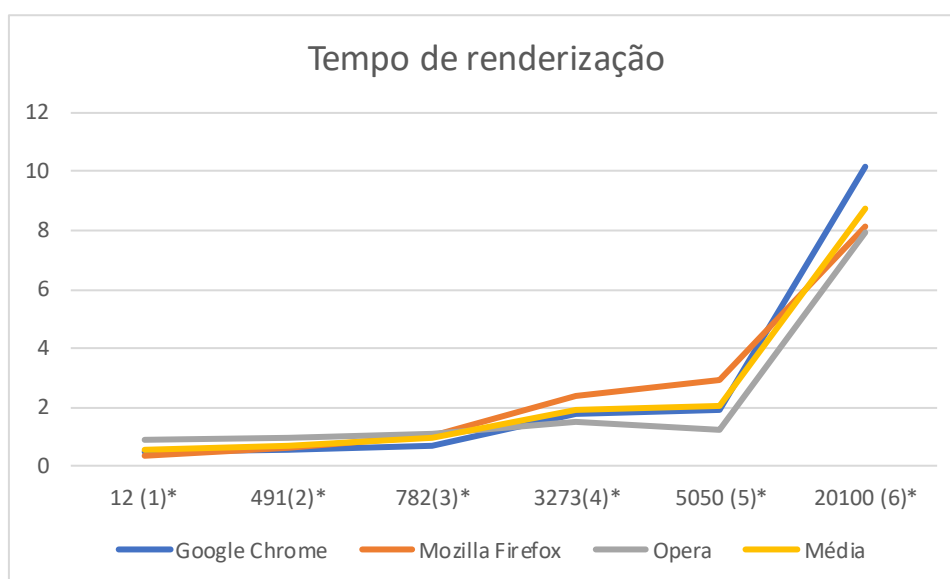


Figura 42 – Gráfico de linhas representando a curva de crescimento (tempo x nº de elementos). Pode-se observar que a plataforma não apresenta variações significativas nos mais diversos navegadores. Fonte: O Autor.

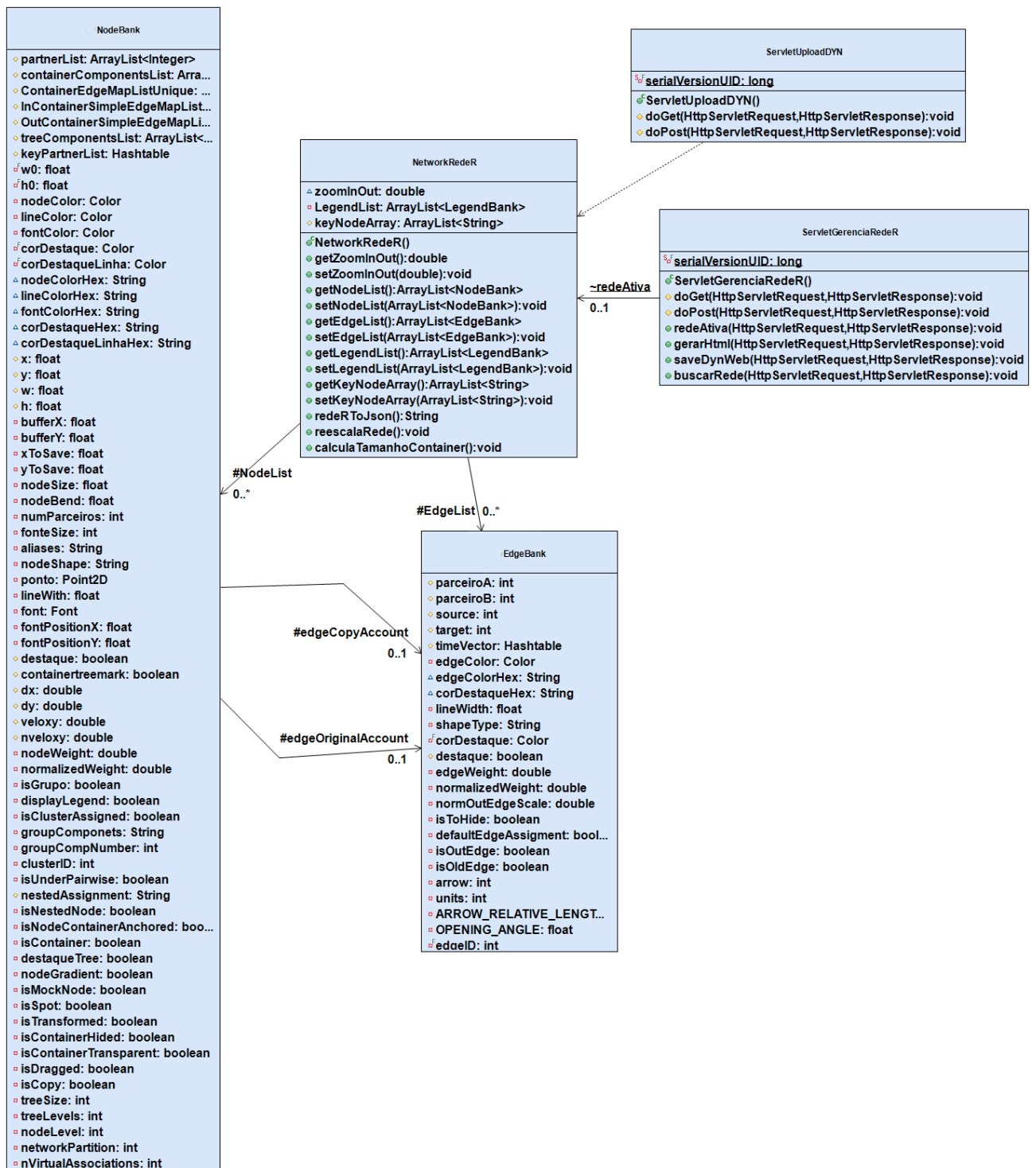


Figura 37 – Diagrama de Classes. Fonte: O Autor.

Capítulo 3



ESTUDO DE CASO

CAPÍTULO 3: ESTUDO DE CASO

1- INTRODUÇÃO

A importância da realização de um estudo de caso para o RedeR Web é questionar a importância da visualização de dados para o entendimento da informação bruta.

Além disso apresentar as dificuldades enfrentados na análise de redes biológicas, confirmando a existência das características hierárquicas e modulares. Podendo então demonstrar em um caso real, com dados do Gene Ontology, a solução promovida pelo RedeR Web.

2- POR QUE O ESTUDO DE CASO COM O GENE ONTOLOGY?

A escolha do Gene Ontology (GO) para a realização do estudo de caso está associada com o objetivo principal do projeto, o modelo de organização estrutural dos dados e sua alta confiabilidade.

O Gene Ontology é um projeto que têm como objetivo reunir informações de genes e produto de genes de uma maneira unificada e padronizada em sua nomenclatura. Entre os dados depositados, há dados de plantas, animais e genomas microbianos, sendo extremamente confiáveis pois há processos de curadoria manual. Dentre suas atividades principais, pode-se encontrar três segmentos:

- 1- Desenvolvimento e manutenção das próprias nomenclaturas e ontologias.
- 2- Anotação de genes e produtos de genes e suas devidas associações.
- 3- Buscar o desenvolvimento de novas ferramentas que otimizem a criação, a manutenção facilitando também o uso das ontologias.

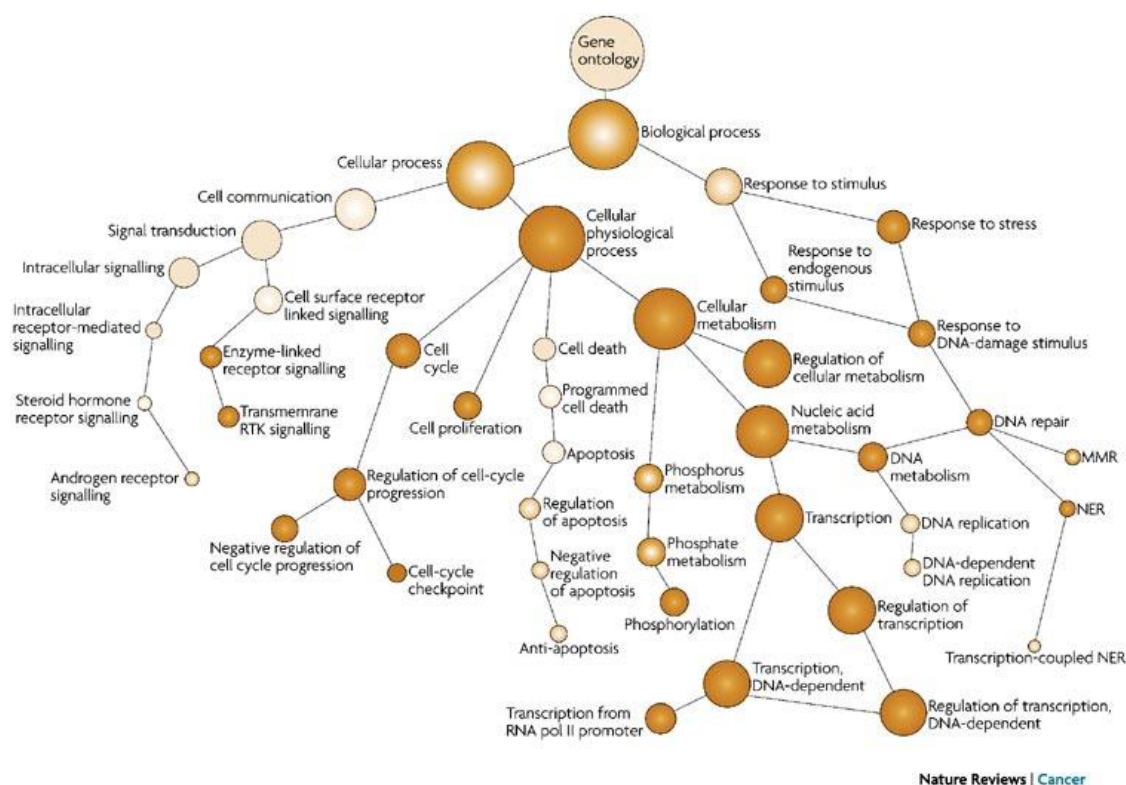


Figura 38 – Estrutura Hierárquica do Gene Ontology:

Fonte: Hu *et al. Nature Reviews Cancer* 7, 23–34 (January 2007) | doi:10.1038 / nrc2036

3- O ESTUDO DE CASO

Para o estudo de caso, optou-se por filtrar do Gene Ontology os termos relacionado com a ontologia de processos biológicos usando dados de anotação do genoma da bactéria *Escherichia coli* strain K12 (Carlson M., 2017) com o pacote disponibilizado no Bioconductor para demonstrar o potencial da plataforma. Para a montagem da rede, também utilizou-se dados provenientes do Gene Ontology, com o pacote Go.db (Carlson, M., 2017) e dados do STRINGdb. (Franceschini, A., 2013)

Ao selecionarmos um termo como raiz e três sub níveis de hierarquias abaixo obtivemos as seguintes estruturas para o termo GO:0050896 (*response to stimulus*):

A presença de 1755 ocorrências de genes mapeados entre os termos e subtermos, com uma redundância de cerca de 46%. Isso deve-se ao fato de que termos podem se repetir em diferentes níveis da hierarquia mapeada no Gene Ontology, como o termo *cellular response to stress* (GO:0033554) que está presente como filho do termo *response to stress* (GO:0006950) e do termo *cellular response to stimulus* (GO:0051716).

É possível perceber que o GO, possui um caráter cíclico entre suas ontologias, sendo necessário o tratamento destes dados para que os genes sejam entidades única na estrutura de dados visando a análise e visualização de uma rede. Para eliminar as replicações de genes, foi criado um script em R, que redefine o pai dos genes com duplicidade dando preferência aos termos em que este possui mais ligações. Sendo assim, a rede formada ficou com 812 genes distintos (nós) e 19.319 interações (arestas).

REPRESENTAÇÃO PLANA DO PROCESSO BIOLÓGICO (RESPONSE TO STIMULUS).

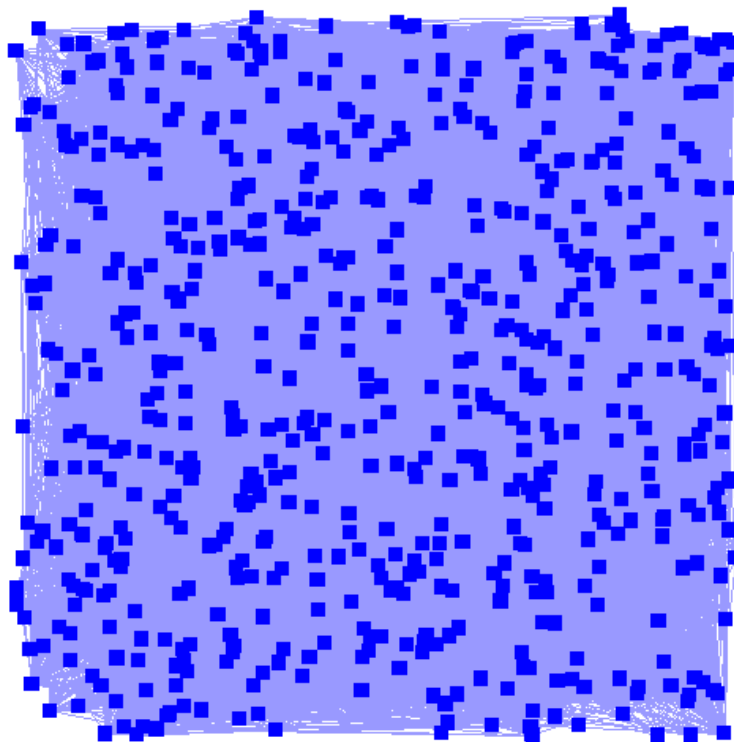


Figura 39 – Representação plana produzida no software Cytoscape 3.5.1 com dados da E. Coli strain K12. Fonte: O autor.

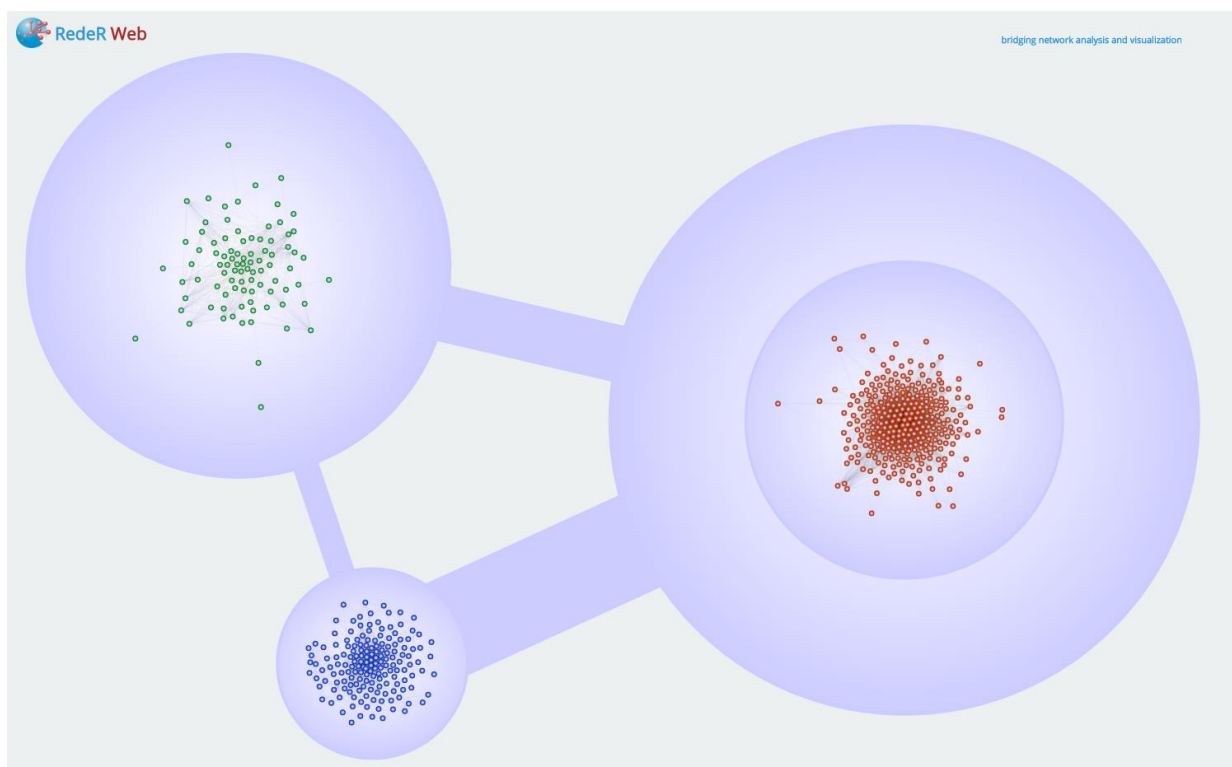


Figura 40 – Representação Modular e Hierárquica produzida no RedeR Web com os dados da E. Coli strain K12. Fonte: O autor.

4- RESULTADOS E DISCUSSÃO

Nota-se que o RedeR Web é capaz de reorganizar uma rede densa obtendo proveito da estrutura hierárquica e modular presente nas informações biológicas. Demonstrando que a ferramenta é estruturalmente e funcionalmente capaz de receber, interpretar e possibilitar a visualização de estruturas hierárquicas comportando o modelo de dados que tem sido amplamente desenvolvido pelo Gene Ontology.

5- CONSIDERAÇÕES FINAIS E PERSPECTIVAS FUTURAS

Neste trabalho introduzimos o RedeR Web, uma plataforma Web que permite ao usuário analisar e visualizar redes densas e hierárquicas com primazia, permitindo que se obtenha conhecimento através dos seus dados brutos, e que isso possa ser compartilhado com outros usuários.

Como o RedeR Web conta com o poder das bibliotecas Javascript, a plataforma permite que o usuário manipule suas redes com ótimo desempenho até mesmo em dispositivos móveis como *tablets* e *smartphones*, sem depender de nenhuma configuração ou instalação de dependências.

Outro aspecto importante, é que não existe a intenção de que a versão web seja substitutiva a versão stand-alone, mas que no RedeR Web seja mantida a compatibilidade entre as versões proporcionando a mesma experiência ao usuário, explorando todos os recursos que uma plataforma no ambiente Web pode oferecer.

Futuramente, queremos aperfeiçoar a plataforma com novos métodos e funções assim como desenvolver uma galeria de redes na qual será possível estabelecer uma comunicação direta entre as versões, e, além disso, estamos estudando a comunicação entre o pacote R e a versão Web.

O RedeR Web além do que já desempenha, apresenta um enorme potencial para ser expandido e se tornar-se a plataforma referência para análise e visualização de redes hierárquicas e modulares, não só para redes biológicas, mas até mesmo para outras áreas do conhecimento.

REFERÊNCIAS

James, J.; Data Never Sleeps 4.0. Disponível em: <https://www.domo.com/blog/data-never-sleeps-4-0/>. Acesso em: 18 nov. 2017

Dumbill, J. Making Sense of Big Data. Big Data. v. 1, p.1-2, 2013.

Laney, D. 3D Data Management: Controlling Data Volume, Velocity, and Variety. Application Delivery Strategies, 2001.

IBM. The four V's of Big Data. Disponível em: <http://www.ibmbigdatahub.com/infographic/four-vs-big-data>. Acesso em: 18 nov. 2017

Demchenko, Y. Big Security for Big Data: Addressing Security Challenges for the Big Data Infrastructure. Proc. Secure Data Management (SMD'13) Workshop. Part of VLDB2013 Conference, p. 26-30, 2013.

Buyya, Rajkumar. Big Data: Principles and Paradigms. Edição 1. Elsevier Inc. 2016.

Marx, V. Biology: The Big challenges of Big Data. Nature. v. 498. p. 255-260, 2013.

EBI. Our Impact. Disponível em: <http://www.ebi.ac.uk/about/our-impact> Acesso em: 18 nov. 2017

Chen, P. C. L. Data-Intensive applications, challenges, techniques and technologies: A Survey on Bi Data. Information Sciences. v. 275, p.314-347, 2014.

Hurwitz. J.; Nugent, A.; Halper. F. Big Data For Dummies. John Wiley & Sons, Inc. 2013.

Kmetová, M. On History of Information Visualization. Distance Learning in Applied Informatics, p.129- 134, 2010.

Aparicio, Manuela; Carlos, C. J. Data Visualization. Communication Design Quarterly. v. 3. p. 7-11, 2014.

Friendly, M & Denis, D. J. Milestones in the history of thematic cartography, statistical graphics, and data visualization, 2009. Disponível em <http://www.math.yorku.ca/SCS/Gallery/milestone/milestone.pdf>. Acesso em: 18 nov. 2017

Card, S. K. The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates, Inc., 1983.

Norretranders, T. The User Illusion: Cutting Consciousness Down to Size. Penguin. Books USA, 1999.

Ware, C. Information Visualization, Third Edition: Perception for Design (Interactive Technologies). Edição 3. Elsevier, Inc. 2013.

Andrews, K. Information Visualisation. Course Notes, 2016. Disponível em: <http://courses.iicm.tugraz.at/ivis/ivis.pdf>. Acesso em: 18 nov. 2017.

Khan, M. & Khan, S. S. Data and Information Visualization Methods, and Interactive Mechanisms: A Survey. International Journal of Computer Applications. v. 34. p. 1-14, 2011.

Mazza, Ricardo. Introduction to Information Visualization. Springer Inc. 2004.

Tufte, R. E. The Visual Display of Quantitative Information. v.13-14. Encyclopedia of mathematics and its applications, 1983.

Aigner, W. et al. Visualization of Time-Oriented Data. Human-Computer Interaction Series. Springer Science & Business Media, 2011.

Card, K. S et al. Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann, 1999.

Gershon, N; Eick, G. S; Card, K. S. Information Visualization. Magazine Interactions. v. 5. p. 9-15, 1998.

Nielsen, J. 2D is Better Than 3D. Disponível em: <https://www.nngroup.com/articles/2d-is-better-than-3d/>. Acesso em: 18 nov. 2017.

Carlson, M. GO.db: A set of annotation maps describing the entire Gene Ontology. Bioconductor. R package version 3.4.1, 2017.

Carlson, M. org.EcK12.eg.db: Genome wide annotation for E coli strain K12. R package version 3.4.1, 2017.

Franceschini, A. et al. "STRING v9.1: protein-protein interaction networks, with increased coverage and integration." Nucleic Acids Research (Database issue), 41, 2013.

Miksch, S. 2D VS 3D. IEEE VIS 2014 Panel Proposal, 2014.

Anscombe, F. J. Graphs in Statistical Analysis. The American Statistician, v. 27. p. 17-21, 1973

Chambers, J. M. Graphical methods for data analysis. Wadsworth International Group. p. 395, 1983

Shneiderman, B. Sparks of Innovation in Human-computer Interaction. Intellect Books, p. 387, 1993

Johnson, B. & Shneiderman, B. Treemaps: A space-filling approach to the visualization of hierarchical information structures. In Proceedings of the 2nd International IEEE Visualization Conference, p. 284-291, 1991.

Wang et al. Visualization of large hierarchical data by circle packing. In Proceedings of the SIGCHI conference on Human Factors in computing systems, p. 517-520, 2006

Heymann JA, et al. Site-specific 3D imaging of cells and tissues with a dual beam microscope. J Struct Biol 155. p. 63-73, 2006

Spence, R. Information Visualization. An Introduction. Edição nº 1, 2001

Bederson, B. B. et. al. A Zooming Web Browser. In Proc. of Multimedia Computing and Networking, v. 2667, p. 260–271, 1996

Leung, Y. K & Apperley, M. D. A review and taxonomy of distortion-oriented presentation techniques. ACM Transactions on Computer-Human Interaction (TOCHI). v.1, p.126-160, 1994

Maletic et al. Visualizing Software for Understanding and Analysis, 2002. Proceedings. First International Workshop. IEEE. p. 32-40, 2002

Boccaletti. S. et al. Complex networks: Structure and dynamics. Physics Reports v. 424, p. 175-308, 2006

Bondy, J. & Murty, U. Graph Theory with Applications. MacMillan, London, 1976

Ruohonen, K., Graph Theory. Loose Leaf, p. 114 2013

Cherifi, H. Complex Networks and Their Applications. Cambridge Scholars Publishing, p. 355, 2014

Metz, J. et al. Redes complexas: conceitos e aplicações. Instituto de Ciências Matemáticas e de Computação, 2007

Amaral, L. A. et al. Classes of small-world networks. Proceedings of the National Academy of Sciences of the United States of America, v. 97, n.21, 2000

Newman, M. E. J. The Structure of Scientific Collaboration Networks. Proceedings of the National Academy of Sciences. p. 404-409, 2001

Barabási & Oltvai, Z. N. Network biology: understanding the cell's functional organization. Nature Reviews Genetics, v.5, p. 101-113, 2004

Watts & Strogatz. Collective dynamics of 'small-world' networks. Nature v.393, p.440-442, 1998

Barabási & Bonabeau. Scale-free Networks. Scientific American. p. 50-59, 2003

Barabási & Albert, R. Emergence of Scaling in Random Networks. Science, v. 286, p. 509-512.

B. Dennis. Molecular Networks: The Top-Down View. Science; Washington. p.1864-1865, 2003.

Albert, R. et al. Error and attack tolerance of complex networks. Nature 406, p. 378-382, 2000

Castro et al. RedeR: R/Bioconductor package for representing modular structures, nested networks and multiple levels of hierarchical associations. Genome Biology 13:R29, 2012

Featherstone, D. & Broadie, K. Wrestling with pleiotropy: Genomic and topological analysis of the yeast gene expression network, v.24, p.267-274, 2002.

Latora, V. & Marchiori, M. Efficient Behavior of Small-World Networks. Physical Review Letters, v. 17 n. 19, 2001

Luo, F. et al 2007. Modular organization of protein interaction networks. Bioinformatics, v. 23, p. 207-214, 2007

Cervone, F. K., Understanding agile project management methods using Scrum. OCLC Systems & Services: International digital library perspectives, v. 27, p. 18-22, 2011

Scrum, J. Scrum: a arte de fazer o dobro do trabalho na metade do tempo. Leya Brasil. p. 240, 2016.

Krasner, G. E. & Pope, S. T. A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System. Parcplace Systems, 1988.

Medvidovic, et al. Modeling Software Architecture in the Unified Modeling Language. ACM Transactions on Software Engineering and Methodology, v. 11, n.1, 2002

Maylor, H. Beyond the Gantt chart: Project management moving on. European Management Journal, v. 19. p. 92-100, 2001

Cervone, F. K., Understanding agile project management methods using Scrum. OCLC Systems & Services: International digital library perspectives, v. 27, p. 18-22, 2011

Scrum, J. Scrum: a arte de fazer o dobro do trabalho na metade do tempo. Leya Brasil. p. 240, 2016.

Krasner, G. E. & Pope, S. T. A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System. Parcplace Systems, 1988.

Medvidovic, et al. Modeling Software Architecture in the Unified Modeling Language. ACM Transactions on Software Engineering and Methodology, v. 11, n.1, 2002

Maylor, H. Beyond the Gantt chart: Project management moving on. European Management Journal, v. 19. p. 92-100, 2001

Pilato, M. C. et al. Version Control with Subversion: Next Generation Open Source Version Control. O'Reilly Media; Edição n. 2, p. 432, 2008

Oracle. Your First Cup: An Introduction to the Java™ EE Platform. Oracle n. 821, 2012

Crockford, D. JSON: The Fat-free Alternative to XML. In Proceedings of XML. Disponível em: <http://www.json.org/fatfree.html>, 2006. Acesso em: 18 nov. 2017